Table of Contents

Executive Summary

Many government and social agencies struggle with determining how to match aid and welfare with the correct households. This is especially prevalent when they aim to target the poorest of populations, as these households have a hard time providing income and education records (along with other forms of necessary data). Therefore, it is important for countries to develop an algorithm that can predict a family's poverty level based on easily measurable characteristics.

Ria, Emma, and Ammar decided to work on this problem because of their past experiences working with nonprofit organizations. They have all worked with education, healthcare, and housing nonprofits domestically and internationally, all of which struggled to determine which areas needed the most funding.

For the purpose of this analysis, we utilized a dataset on Costa Rican households. This dataset comes from the Inter-American Development Bank, which works on providing financing in Latin America and the Carribean. Costa Rica presents an interesting dilemma. Despite being one of Latin America's and the Carribean's biggest social spender, their poverty rate has remained largely unchanged over the past twenty years. To amend this problem, in the analysis below, we have developed a model that predicts the poverty level that a household falls into with **74.75% accuracy**. The model can serve as a starting point for countries (not just Costa Rica) to determine what characteristics are most indicative of certain poverty levels and increase accuracy with addressing a household's social need.

Data Overview

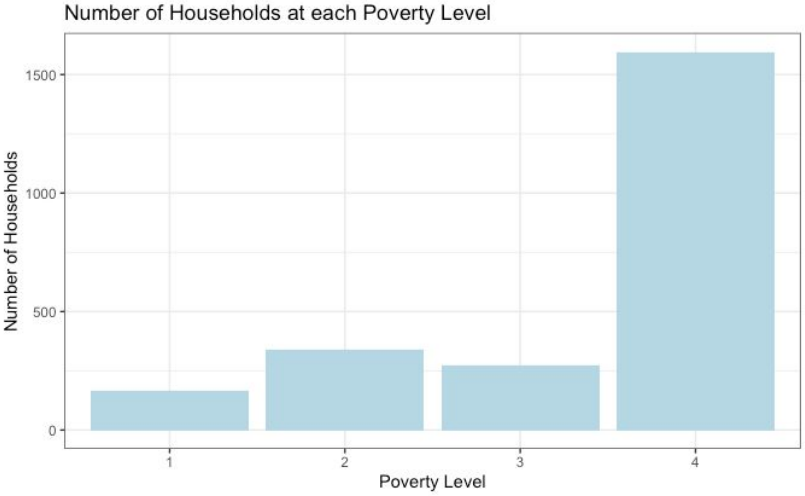
We utilize a dataset of 7606 observations, each of which represents an individual. There are a total of 142 predictors. Each individual is part of a household, so he/she has both an individual ID and a household ID. Each individual also has 1 of 4 *Target Poverty Levels* (every individual in a household has the same *Target Poverty Level*). Below is a summary of the response variable:

- 1: Extreme Poverty
- 2: Moderate Poverty
- 3: Vulnerable Households
- 4: Non-Vulnerable Households

One way of segmenting the predictors is into individual and household level variables. Examples of individual variables include maximum level of education attained, marital status, and gender. Examples of household variables include types of building material, sewage situation, energy sources, and number of phones and tablets in the house.

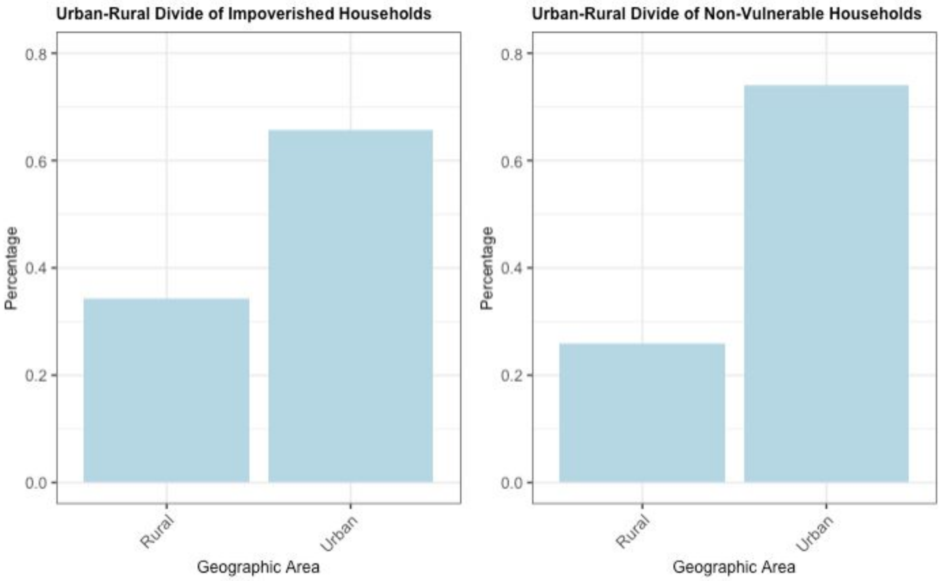
Exploratory Analysis

First, we look at the *distribution of households by poverty level*. This dataset is very imbalanced, with the vast majority of households being non-vulnerable. This problem could be tackled through oversampling, but for now, we will leave that technique out of this project due to time constraints.



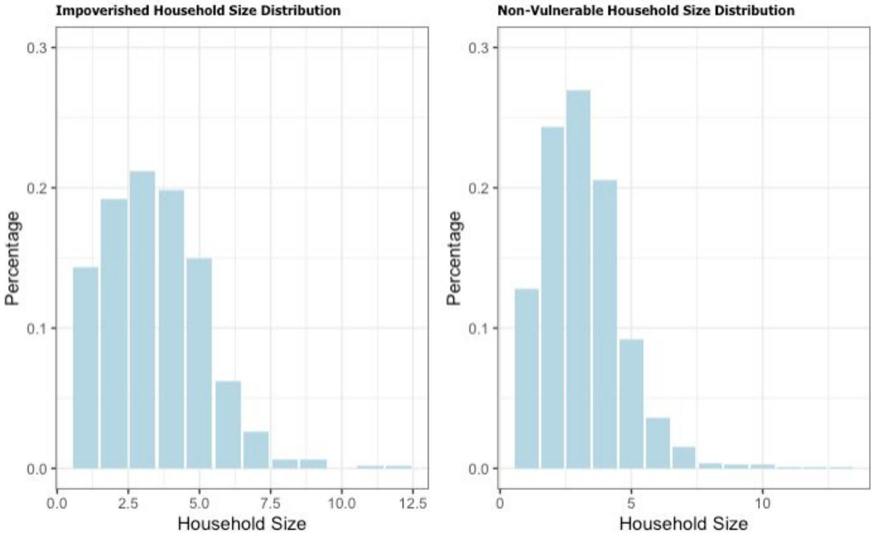
In the histogram pictured above, we can see that the number of households in each poverty level are very skewed. There are a significant number of households in poverty level 4, while there is a lot less in levels 1, 2, and 3. This means that in the data we are examining, there is a high number of households who are not in poverty (i.e. poverty level 3 or 4). **For the rest of the EDA and model building moving forwards, we have decided to transform *Poverty* to be a binary variable and categorize poverty levels 1 and 2 as "1" or impoverished, while poverty levels 3 and 4 are considered "0" or non-vulnerable.**

We start our EDA looking big picture at poverty levels in *urban vs. rural areas*.



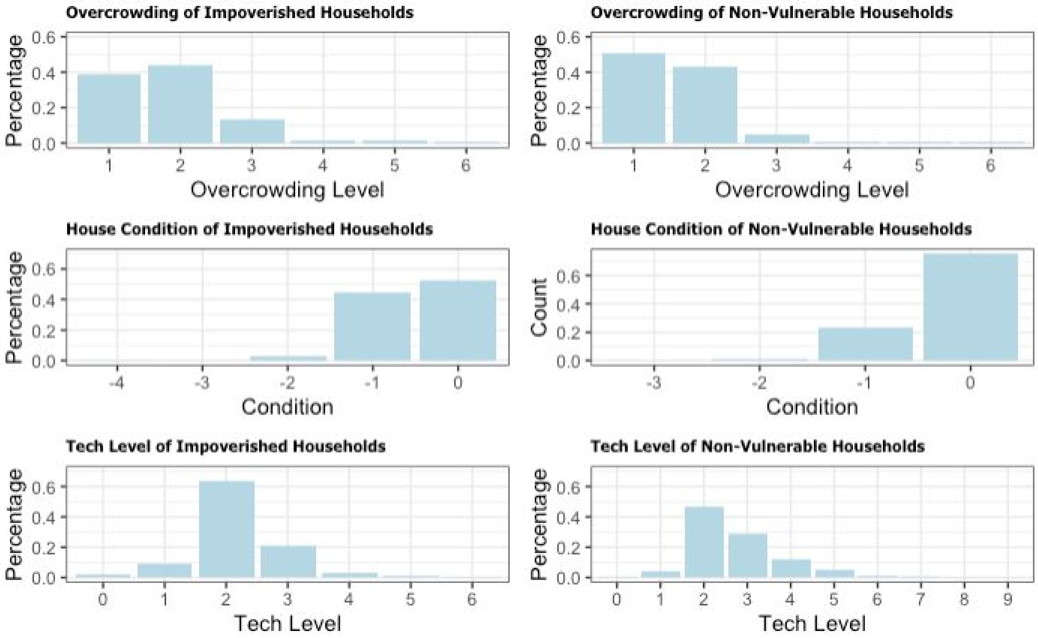
In the histograms above, we see that a higher proportion of impoverished households (~35%) live in rural areas compared to non-vulnerable households (~22%). This is expected, considering there are better jobs and quality of life in urban areas, resulting in a cost of living that poorer households could not afford.

Next, we look into households specifically, starting with *household size*.



In the graphs above, ~70% of non-vulnerable households are made up of 2 to 4 people, while only ~58% of impoverished households are made up of 2 to 4 people. A higher percentage of impoverished households have large household sizes, a trend that we expected to see.

Next, we look further into specific households, specifically at *overcrowding* (# of people per room), *overall house condition*, and *amount of technology* in the home.

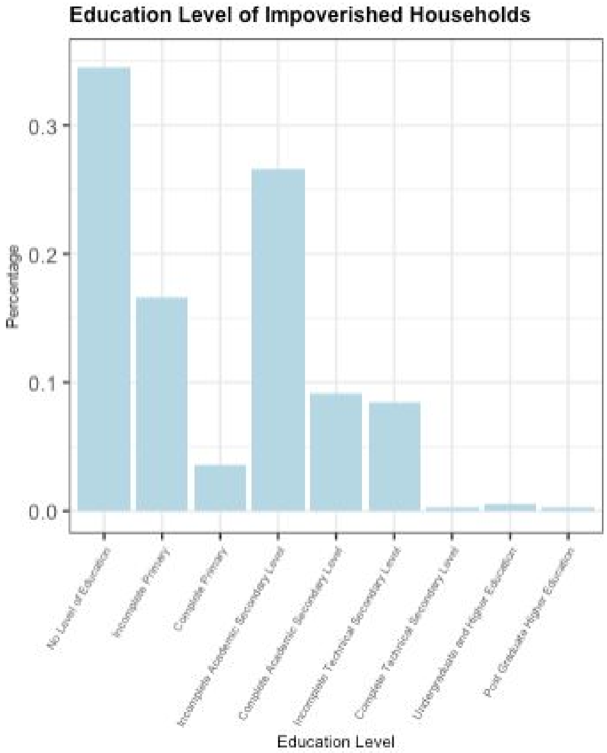


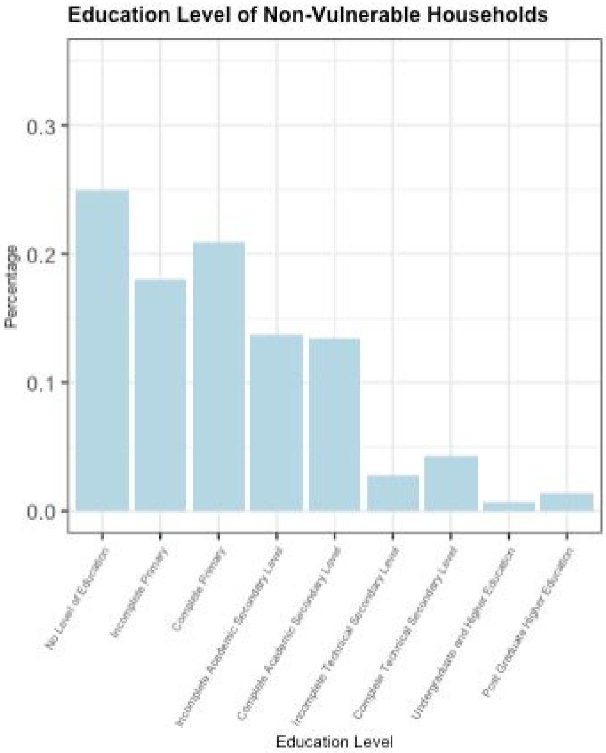
For overcrowding, it is expected that impoverished households would have more people per room, with ~50% of them having 2-3 people per room compared to ~40% of non-vulnerable households having 2-3 people per room. This is not a large difference, but could be significant.

For household condition, the more negative the condition is, the greater the lack of necessities like toilet, water, a roof, etc. Condition was a variable that we created for more concise EDA and model analysis. As pictured, it is expected that non-vulnerable households will have better living conditions meaning a condition value closer to 1, while impoverished households typically have a greater negative condition value.

For tech level, the graphs shows that the mode of tech level of impoverished households is 2. It is evident that the tech level of non-vulnerable households is more widely distributed, ranging from 1 to 9. Intuitively, this makes sense as non-vulnerable households should have more disposable income to spend on technology. This is also a variable we created.

Next, we look at the people in the households, specifically the *education level of the head of household*.



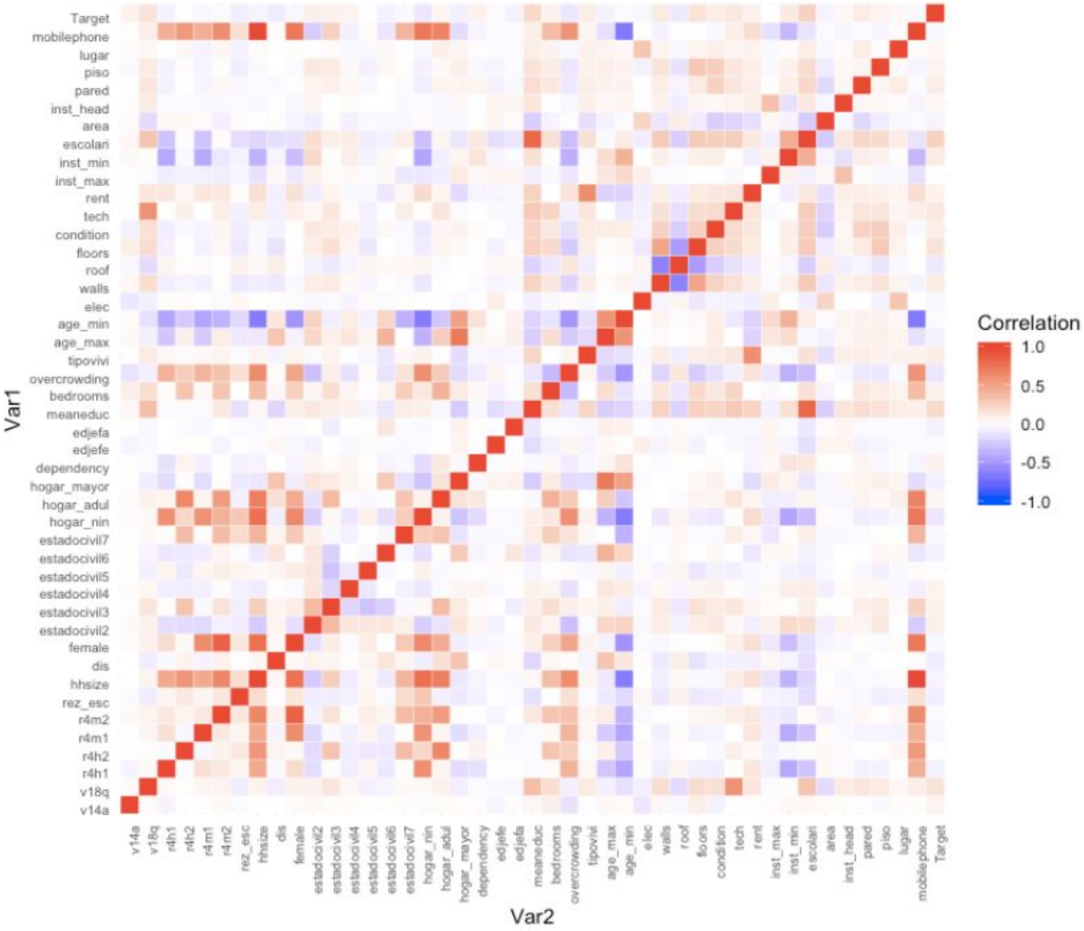


It is expected that impoverished households would have a higher percentage of *No level of Education*; however, it is quite surprising that there is also a very high percentage of non-vulnerable household heads with household head having *No level of Education*. This could be because Costa Rica is a country with a relatively lower education rate overall. There is also much more consistency in the education level of non-vulnerable households. In the impoverished household histogram, we can also see that it is bi-modal--there are two evident peaks at *No level of Education* and *Incomplete Academic Secondary Level*. This could be due to the fact that many people who are the heads of their impoverished household had to quit their secondary education to support their families.

Feature Engineering

After our EDA, we realized that we have a lot of predictors that could be correlated and needed to be engineered to be able to be run in a model. Therefore, we did the following.

Removed Correlated Predictors



Based on the correlation matrix above, we can see the following.

- **age_min** and **age_max** are correlated with a lot of predictors, so we remove them
- **r4h1, r4h2, r4m1**, and **r4m2** (signify numbers of males and females of different ages in the house) are all pretty correlated with the **hogar** variables. We keep the **hogar** variables because they are less correlated with other predictors overall.
- **tech** and **v18q** are correlated because **v18q** signifies owning a tablet. We remove **v18q**.
- **escolari** and **mean_educ** are correlated. We remove **escolari**.
- **inst_min** and **inst_max** are correlated. We remove **inst_min**.

Created Ordinal Variables

Many of the predictors have an inherent ordering. Below is a list of the predictors that were converted into factors, with 1 representing the worst level of the predictor.

- **elec** (source of electricity)
- **wall** (wall quality - eg. good, bad, regular)
- **roof** (roof quality)
- **floor** (floor quality)
- **inst** (maximum level of education - eg. high school)
- **escolari** (years of school)
- **pared** (wall material)
- **piso** (floor material)
- **techo** (roof material)
- **abasta** (location of water source)
- **sanitario** (sewage system)
- **energcocinar** (energy source)
- **elimbasu** (trash disposal system)
- **tipovivi** (ownership/rent level of house)

Created New Features

- **condition** (more negative with more lack of necessities like toilet, electricity, etc)
- **tech** (more positive with more discretionary tech owned like tablets)
- **rooms** (standardized based on number of people in the house)
- **rent** (standardized based on number of rooms in the house)

Finally, all of the household features were aggregated through either taking the sum or mean of individual's features. For example, **bedrooms** were taken as an average of all the individuals in the house because every individual would have the same number of bedrooms if they lived in the same house. **mobilephone**, on the other hand, was summed because we cared more about the total number of phones owned by household.

PCA

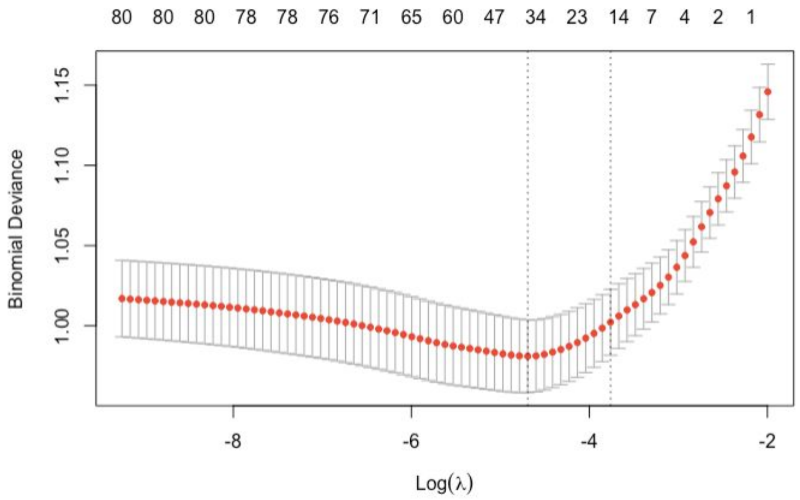
We did not utilize PCA because our dataset did not have enough numerical predictors. Utilizing PCA would have resulted in too many principal components, with none explaining enough of the variance. (See appendix)

Model Building

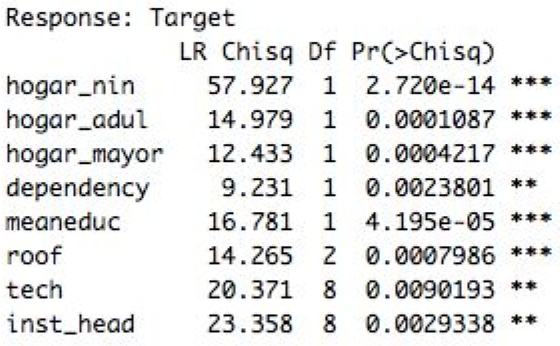
We start by splitting the final cleaned dataset, using 75% for training and 25% for testing. In addition, we decided to make **Target** a binary variable (1 signifies at most risk of poverty and 0 signifies at least risk of poverty) for this analysis. From the above designations (in *Data Overview*), levels 1 and 2 are assigned to 1 and levels 3 and 4 are assigned to 0.

LASSO

Since we end up with 36 predictors after EDA, we start with LASSO in order to have a more parsimonious set of predictors.



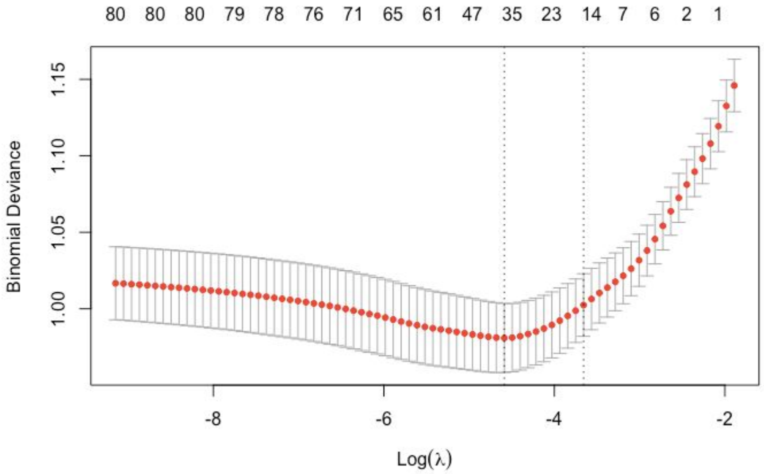
Here, **lambda.min** would provide us with a model of ~35 predictors, while **lambda.1se** would result in a model with ~16 predictors. We use **lambda.1se** because this would be a much more parsimonious model. Then, we perform backward selection, kicking out predictors that are insignificant one at a time until all predictors are significant at α = 0.01 level. Below is the final output from LASSO. From that output we can see that the LASSO predictions emphasized the amount of people in the home through **hogar_nin**, **hogar_adul**, **hogar_mayor**, **dependency** which all were variants on the different individuals in the home. In addition, home factors such as the quality of the **roof** and the **tech** in the home were predictive as well. Lastly, the education of both the head of house, **inst_head**, and the household average, **meaneduc**, were important in predicting poverty. Ultimately, our **LASSO model predicted poverty with 75.5% accuracy**.



```
Response: Target
            LR Chisq Df Pr(>Chisq)
hogar_nin     57.927  1  2.720e-14 ***
hogar_adul    14.979  1  0.0001087 ***
hogar_mayor   12.433  1  0.0004217 ***
dependency     9.231  1  0.0023801 **
meaneduc      16.781  1  4.195e-05 ***
roof          14.265  2  0.0007986 ***
tech          20.371  8  0.0090193 **
inst_head     23.358  8  0.0029338 **
```

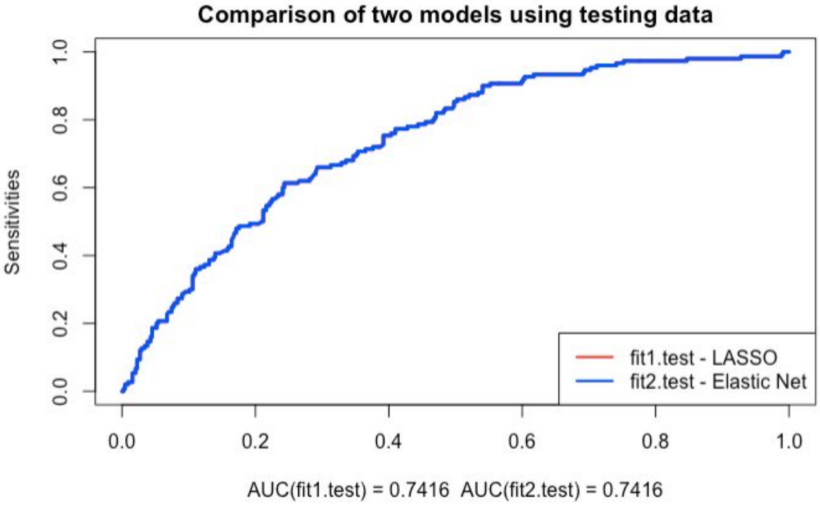
Elastic Net

We also consider a few elastic net models, ultimately settling on an alpha of 0.90. Elastic Net is supposed to overcome 2 limitations: LASSO doesn't perform grouped selection, so it tends to select one variable from a group of very correlated variables and ignore the others and having more predictors (p) than data points (n), LASSO will only select n predictors at most.



The output model is the same as LASSO, as shown through the plot.

LASSO and Elastic Net Evaluation



In this case, LASSO and Elastic Net gave us the same model. This model has an **AUC of 0.7416**, which is satisfactory.

In an attempt to get a higher AUC, we tried the following:

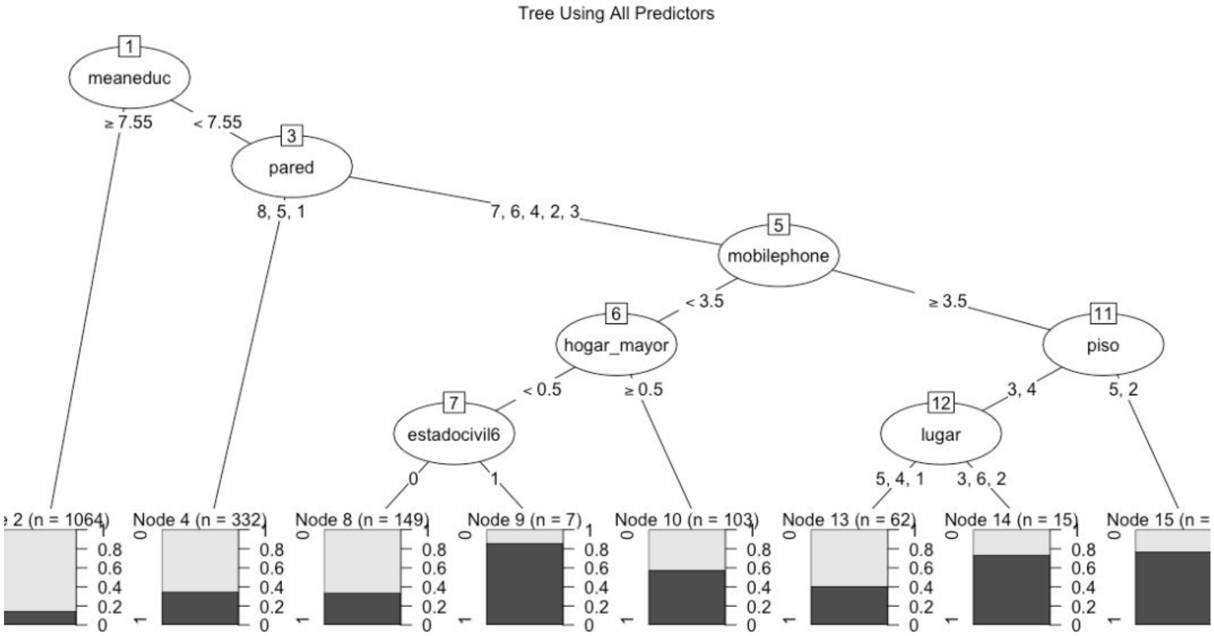
- Changing the number of correlated predictors that were removed before creating the training/testing sets
- Changing the alpha for the Elastic Net model
- Changing the significance threshold for our final model (from 0.01 to 0.05)

None of the following changed the AUC by a significant amount, so we chose to stick with this.

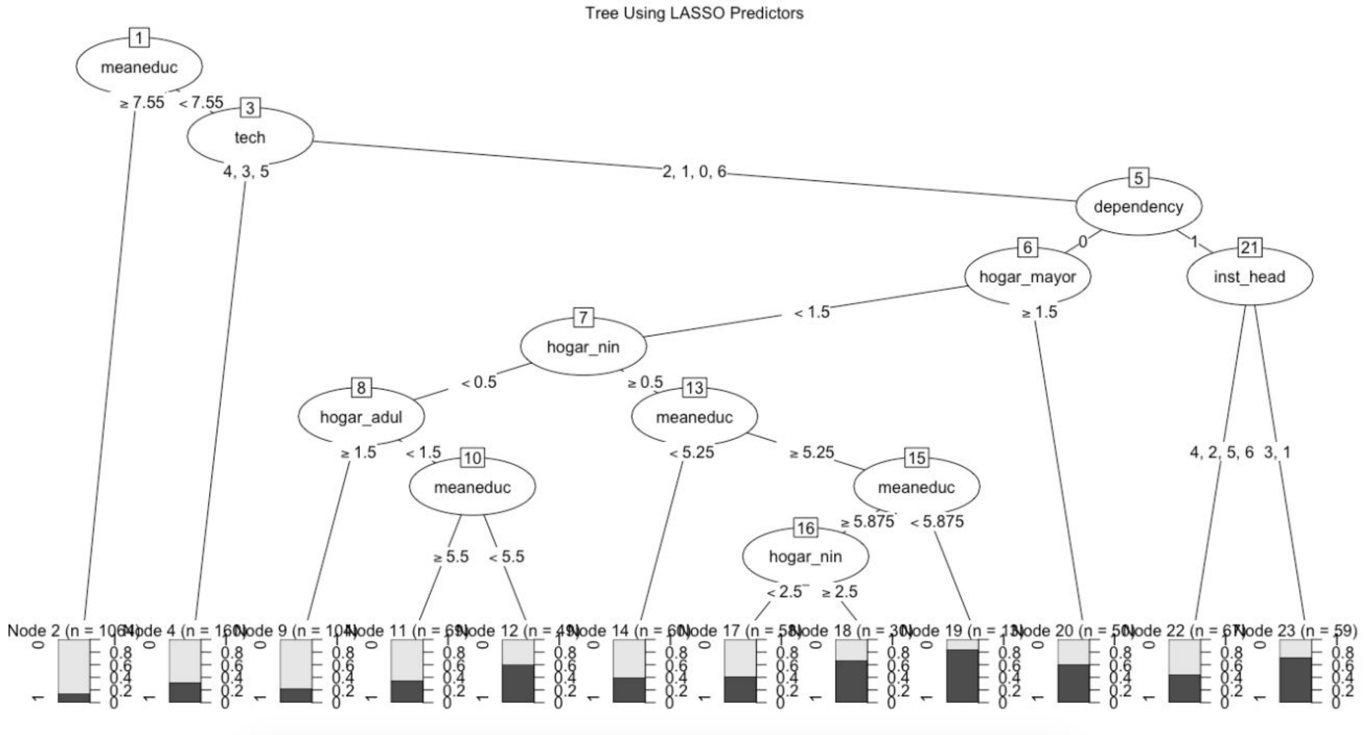
Decision Tree

Decision trees are another type of classifier that we decided to use to build a model for our data. In this scenario we thought it would be a good comparison to make against the Random Forest model, as they are both related. Decision trees are advantageous because they are not a linear model, so there is greater flexibility in the classification. Additionally, they take interactions among variables into account and are easy to interpret. On the other hand, decision trees are greedy and forward splitting, which is not optimal; they are also not stable and often suffer from overfitting, which results in a bad prediction, and possibly very high testing error when compared to the training error.

We first fit a decision tree using all of the predictors:



Next, we fit a decision tree using only the predictors outputted by LASSO/Elastic Net:

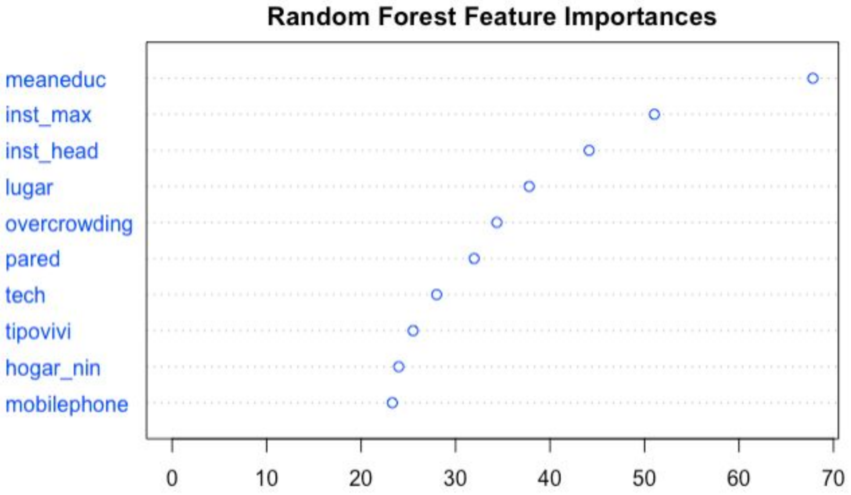


Comparing the two trees we see there is little different in accuracy, with the **LASSO tree having 72.06% accuracy** while the **random tree had 74.08% accuracy**. In addition when looking at the tree, they have similar structures with slight differences. Both trees evaluate education through the **mean_educ** variable, the demographic of individuals in the household with **hogar_** variables. The one main difference between the trees is their evaluation of **tech** and household amenities. The first tree found the **mobilephone**, **piso**, and **lugar** variables to all be telling in evaluating incidences of poverty.

Random Forest

Random forest builds upon the weaknesses of decision trees in that it is an ensemble learning algorithm. Random forests can be tuned better to avoid overfitting the data, which is a key risk in decision trees, and they typically have lower bias and lower variance when compared to normal decision trees. Random forests are able to have lower variance through bagging, using multiple bootstrapped samples. Thus, Random Forests are able to reduce bias. They also can be trained quickly and can handle large numbers of predictors. We ran the random forest on all the variables in the training set. However, in order to generate a parsimonious forest, we tuned the forest to find the optimal number of splits, which was 4 (see appendix). In addition, we tuned the model to find the optimal number of trees for the model at 250. Ultimately, our Random Forest model was **accurate in predicting poverty 74.75% of the time**, in line with our other models. However, Random Forest gave us a different perspective on the predictors that were most predictive. It put a much higher importance on education level, with three most important predictors **meaneduc**, **inst_max**, **inst_head**, which are all related to the education levels of the house. From there, the Random Forest selected variables that looked at the family's housing condition, looking at which region it was located, **lugar**, how overcrowded the house was, **overcrowding**, and the quality of walls and tech in the house, **pared** and **tech**.

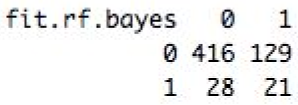
Feature Importances



Model Evaluation

Model	Accuracy
LASSO/Elastic Net	75.5%
Decision Tree (All Predictors)	74.08%
Decision Tree (LASSO/Elastic Net Predictors)	72.06%
Random Forest	74.75%

Confusion Matrix for Random Forest



```
fit.rf.bayes   0   1
           0 416 129
           1  28  21
```

We have **Precision = 76.33%, Recall = 93.69%, F1 Score = 84.13%**. This shows that ~76% of our results are relevant. 93.69% of those relevant results are classified correctly. The model does a decent job of minimizing false negatives (good, because we want to minimize the number of times the model does not detect a household with poverty). The F1-Score is a combination of precision and recall. **F1-Score is a better measure than accuracy for imbalanced classes (a situation we have)**, and since .8413 is pretty close to 1, we would deem these to be good results.

Conclusion

Despite all of the models have similar accuracy, the Random Forest model should be selected for classifying impoverished and non-vulnerable households for the reasons listed below:

1. The random forest model has significant predictive power, when compared to logistic regressions and the decision tree. The AUC of the random forest is .744 (as shown in the appendix 5) vs. the AUC of the LASSO and elastic net model of .741.
2. The Random Forest model is quite interpretable because individual features and values can be highlighted.
3. In this scenario, the Random Forest model intuitively works better for this classification problem because classifying poverty level has a nonlinear boundary as opposed to LASSO, which works better for classifying with linear boundaries.

In this prediction, two of the most important variables were education and location variables. In order to finally ameliorate it's poverty rate, Costa Rica should invest in education in the most distressed areas such as Central. It seems it should target this education in primary education. As seen earlier, there is a large dropoff between the impoverished and non-vulnerable group in if they get any education, and if they complete that basic education. Before investing in higher education, Costa Rica needs to get these students into school in the first place. Moreover, early childhood education programs in America, such as the Perry Preschool project, had returns that served these children for a lifetime with just a single year of preschool.

The second most important set of predictors were those related to housing. Costa Rica has a low homelessness rate; however it has a high home insecurity rate at 52%, which our predictor supports. It found that the existence of basic home essentials such as quality of walls was predictive in finding poverty. Additionally, homes that had more overcrowding were more likely to be impoverished. Investing in housing development, would then be a very wise investment in Costa Rica for several reasons. First, increasing the housing stock would decrease the price in the market for everyone. Secondly, improving the housing conditions for those in poverty helps protect these individuals from adverse weather conditions, sickness, and allievates these individual's tight family budgets thereby improving both physical and mental health.

Ultimately, we hope this project is helpful in developing algorithms that enables governments and social organizations to better understand how to determine poverty levels. Additionally, we believe that the features identified throughout the model building process showcase important attributes that signal poverty that will allow organizations to improve localities in a specific manner.

Extensions and Limitations

While we feel that our model paints an accurate picture of capturing the incidence of poverty in Costa Rica, there are a few ways our analysis could be improved. While our dataset contained information on education and living conditions, it did not contain data on health and access to food. In addition, there was no individual data on employment in homes. Having food to put on the table every day and a job to provide for a family, are critical indicators of life and poverty and would improve both the accuracy of the model and its descriptiveness.

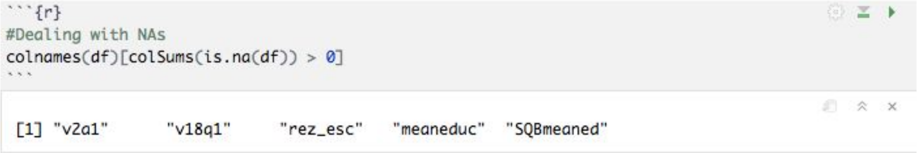
Secondly, poverty cutoffs used by the Inter-American Bank is based on the global poverty line of $1.25. However, poverty isn't just a numbers game, it, unfortunately, is a lifestyle, being one cent above or below this line doesn't materially change someone's life, but for the purposes of the Bank's classification, those people are in different groups. In addition, data gathering efforts from government institutions historically underestimate poverty. This is because it is hard to count people in poverty and find them, due to the housing insecurity these individuals face.

Lastly, due to constraints in time and ease of use, we used a binary regression rather than a multinomial regression. However, in order to design policy interventions, there is a difference in helping those in extreme poverty versus people just under the poverty line.

Yet, despite these hurdles we made a prediction model with 75%, which is very valuable. This model could be used not just on Costa Rica, but as a tool for other Latin American and Caribbean countries that have grappled with similar issues. Not only could governments use this information, but NGOs could better target their spend and appeal to donors about how much further their dollars will go.

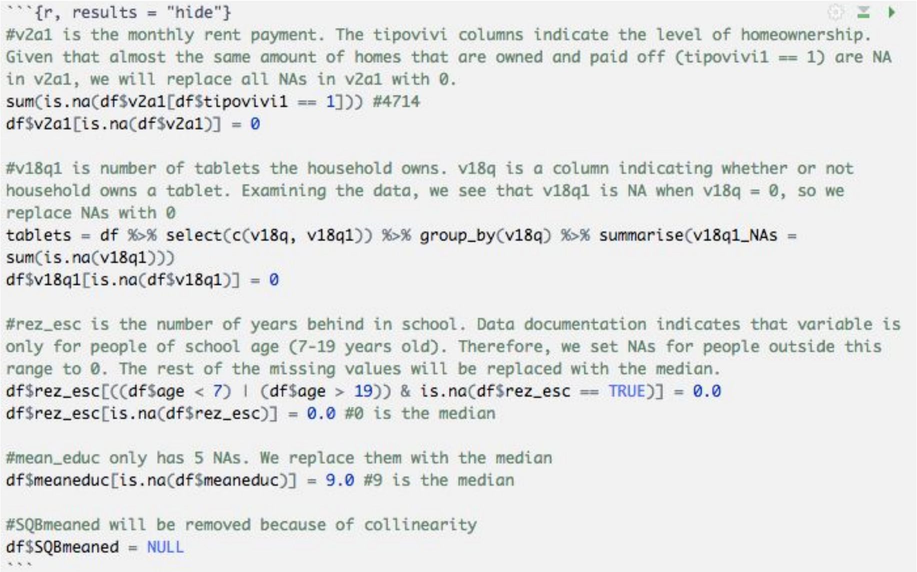
Appendix

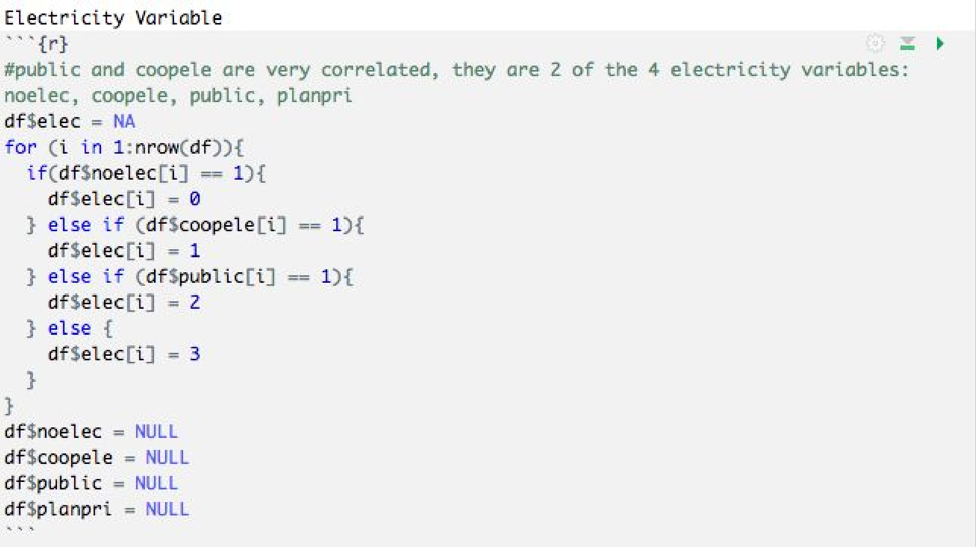
1. Dealing with NAs



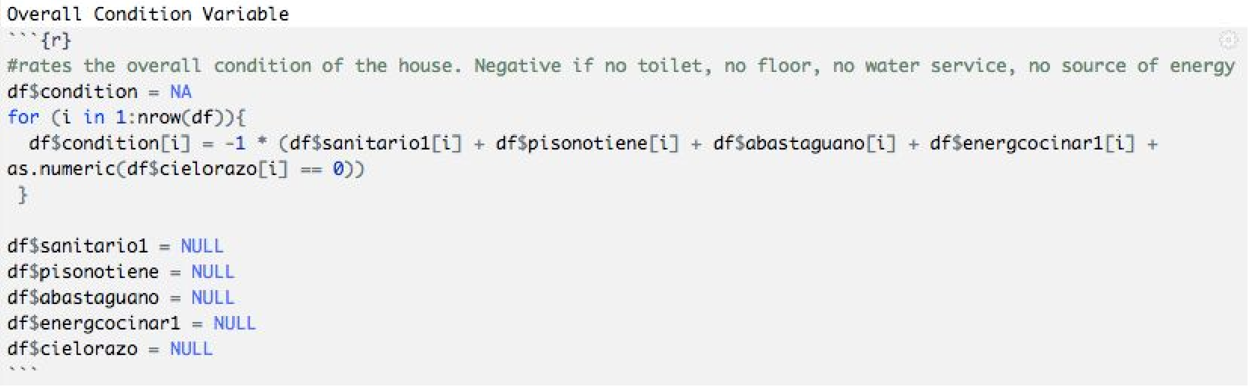
~~~
```{r}
#Dealing with NAs
colnames(df)[colSums(is.na(df)) > 0]
```
~~~

```
[1] "v2a1"      "v18q1"      "rez_esc"      "meaneduc"    "SQBmeaned"
```

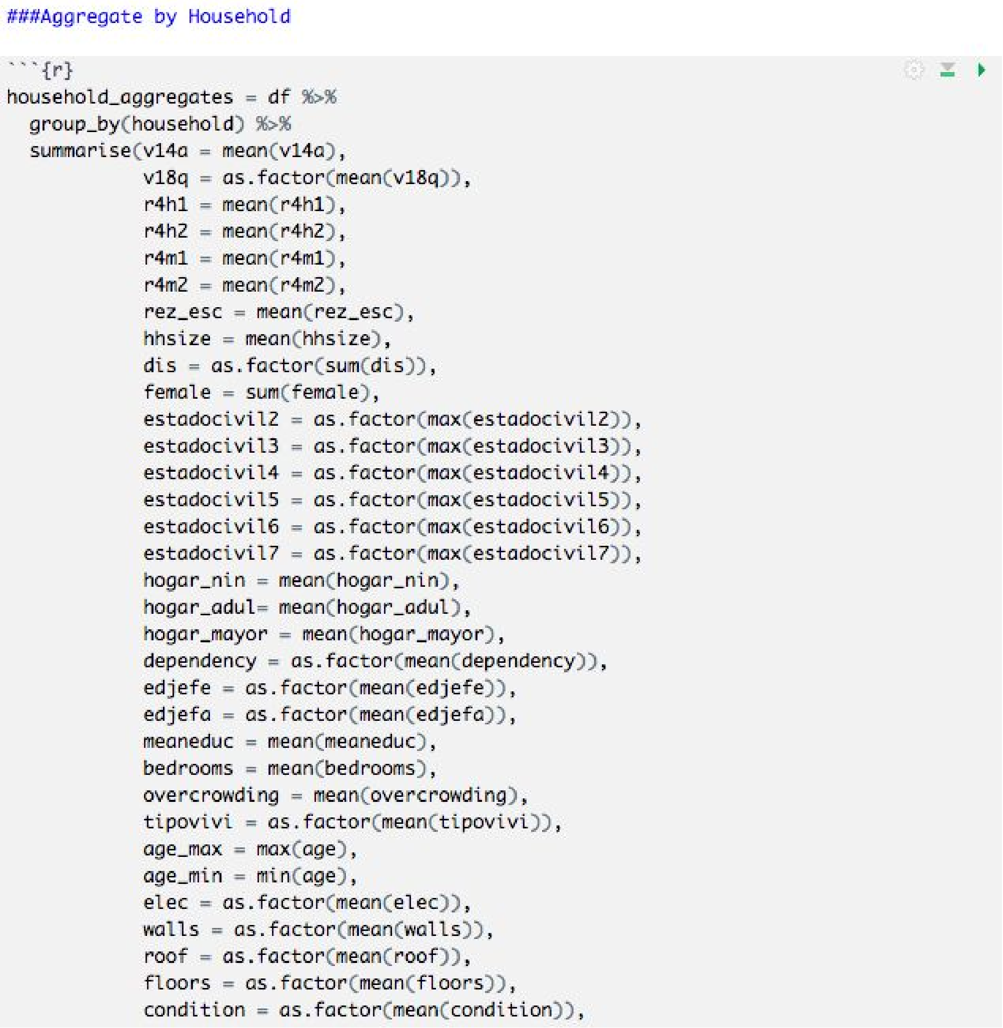


~~~
```{r, results = "hide"}
#v2a1 is the monthly rent payment. The tipovivi columns indicate the level of homeownership.
Given that almost the same amount of homes that are owned and paid off (tipovivi1 == 1) are NA
in v2a1, we will replace all NAs in v2a1 with 0.
sum(is.na(df$v2a1[df$tipovivi1 == 1])) #4714
df$v2a1[is.na(df$v2a1)] = 0

#v18q1 is number of tablets the household owns. v18q is a column indicating whether or not
household owns a tablet. Examining the data, we see that v18q1 is NA when v18q = 0, so we
replace NAs with 0
tablets = df %>% select(c(v18q, v18q1)) %>% group_by(v18q) %>% summarise(v18q1_NAs =
sum(is.na(v18q1)))
df$v18q1[is.na(df$v18q1)] = 0

#rez_esc is the number of years behind in school. Data documentation indicates that variable is
only for people of school age (7-19 years old). Therefore, we set NAs for people outside this
range to 0. The rest of the missing values will be replaced with the median.
df$rez_esc[((df$age < 7) | (df$age > 19)) & is.na(df$rez_esc == TRUE)] = 0.0
df$rez_esc[is.na(df$rez_esc)] = 0.0 #0 is the median

#mean_educ only has 5 NAs. We replace them with the median
df$meaneduc[is.na(df$meaneduc)] = 9.0 #9 is the median

#SQBmeaned will be removed because of collinearity
df$SQBmeaned = NULL
~~~
```

2. Examples of Ordinal Variable Creation



```
Electricity Variable
```{r}
#public and coopele are very correlated, they are 2 of the 4 electricity variables:
noelec, coopele, public, planpri
df$elec = NA
for (i in 1:nrow(df)){
  if(df$noelec[i] == 1){
    df$elec[i] = 0
  } else if (df$coopele[i] == 1){
    df$elec[i] = 1
  } else if (df$public[i] == 1){
    df$elec[i] = 2
  } else {
    df$elec[i] = 3
  }
}
df$noelec = NULL
df$coopele = NULL
df$public = NULL
df$planpri = NULL
```
```
```

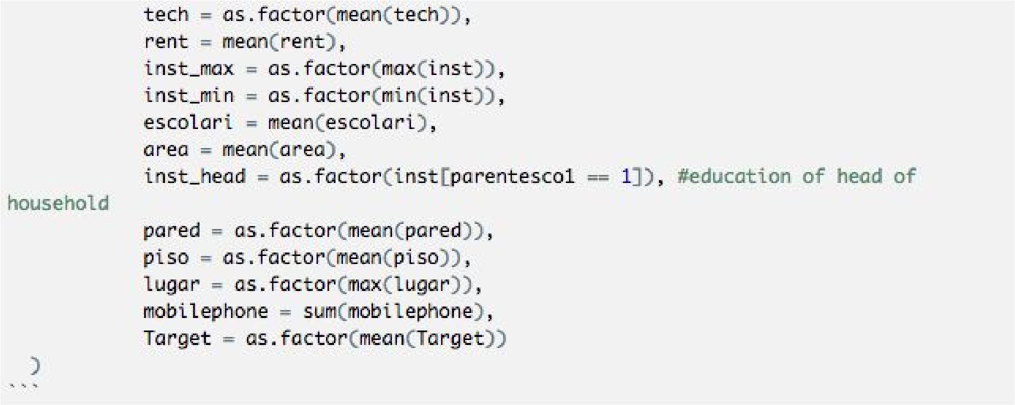
3. Overall Condition Variable

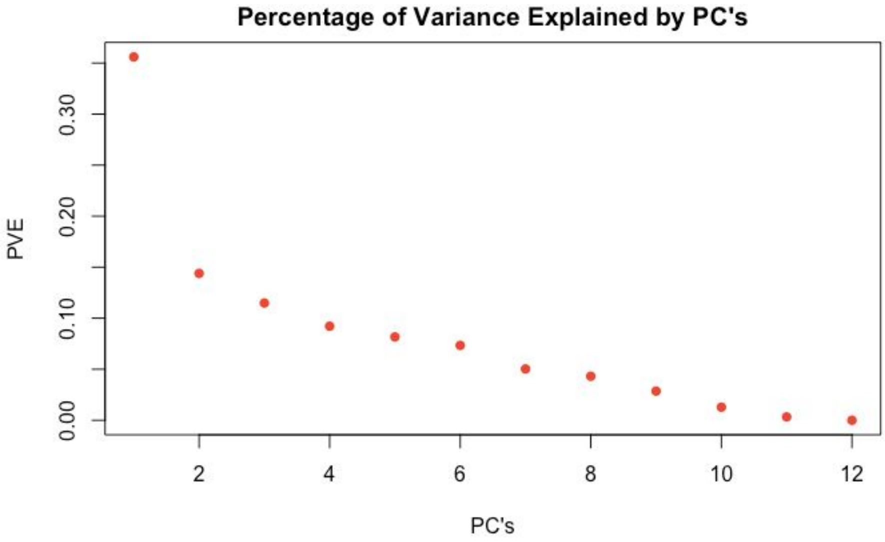
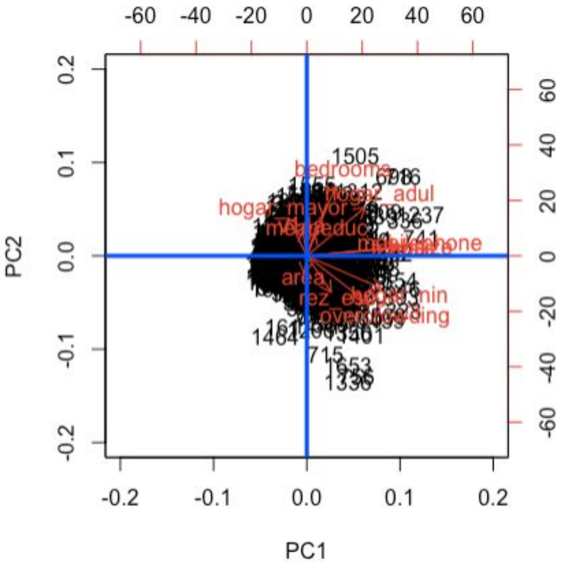


Overall Condition Variable

~~~
```{r}
#rates the overall condition of the house. Negative if no toilet, no floor, no water service, no source of energy
df$condition = NA
for (i in 1:nrow(df)){
  df$condition[i] = -1 * (df$sanitario1[i] + df$pisonotiene[i] + df$abastaguano[i] + df$energcocinar1[i] +
as.numeric(df$cielorazo[i] == 0))
 }

df$sanitario1 = NULL
df$pisonotiene = NULL
df$abastaguano = NULL
df$energcocinar1 = NULL
df$cielorazo = NULL
~~~
```

## 4. Aggregation of predictors from individual to household level



```
###Aggregate by Household

```{r}
household_aggregates = df %>%
  group_by(household) %>%
  summarise(v14a = mean(v14a),
            v18q = as.factor(mean(v18q)),
            r4h1 = mean(r4h1),
            r4h2 = mean(r4h2),
            r4m1 = mean(r4m1),
            r4m2 = mean(r4m2),
            rez_esc = mean(rez_esc),
            hhsize = mean(hhsize),
            dis = as.factor(sum(dis)),
            female = sum(female),
            estadocivil2 = as.factor(max(estadocivil2)),
            estadocivil3 = as.factor(max(estadocivil3)),
            estadocivil4 = as.factor(max(estadocivil4)),
            estadocivil5 = as.factor(max(estadocivil5)),
            estadocivil6 = as.factor(max(estadocivil6)),
            estadocivil7 = as.factor(max(estadocivil7)),
            hogar_nin = mean(hogar_nin),
            hogar_adul= mean(hogar_adul),
            hogar_mayor = mean(hogar_mayor),
            dependency = as.factor(mean(dependency)),
            edjefe = as.factor(mean(edjefe)),
            edjefa = as.factor(mean(edjefa)),
            meaneduc = mean(meaneduc),
            bedrooms = mean(bedrooms),
            overcrowding = mean(overcrowding),
            tipovivi = as.factor(mean(tipovivi)),
            age_max = max(age),
            age_min = min(age),
            elec = as.factor(mean(elec)),
            walls = as.factor(mean(walls)),
            roof = as.factor(mean(roof)),
            floors = as.factor(mean(floors)),
            condition = as.factor(mean(condition)),
```

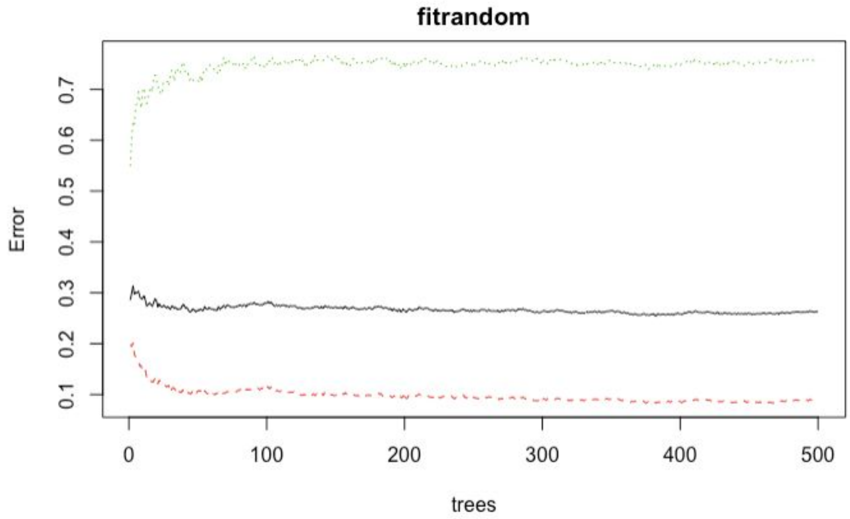


```
            tech = as.factor(mean(tech)),
            rent = mean(rent),
            inst_max = as.factor(max(inst)),
            inst_min = as.factor(min(inst)),
            escolari = mean(escolari),
            area = mean(area),
            inst_head = as.factor(inst[parentesco1 == 1]), #education of head of
household
            pared = as.factor(mean(pared)),
            piso = as.factor(mean(piso)),
            lugar = as.factor(max(lugar)),
            mobilephone = sum(mobilephone),
            Target = as.factor(mean(Target))
  )
```
```

## 5. Visualization of PCA application




```

As mentioned above, this is the PCA exploratory analysis with the numerical predictors in our dataset. Only ~¼ of our data was numerical, so we decided that PCA would not be useful for this analysis.

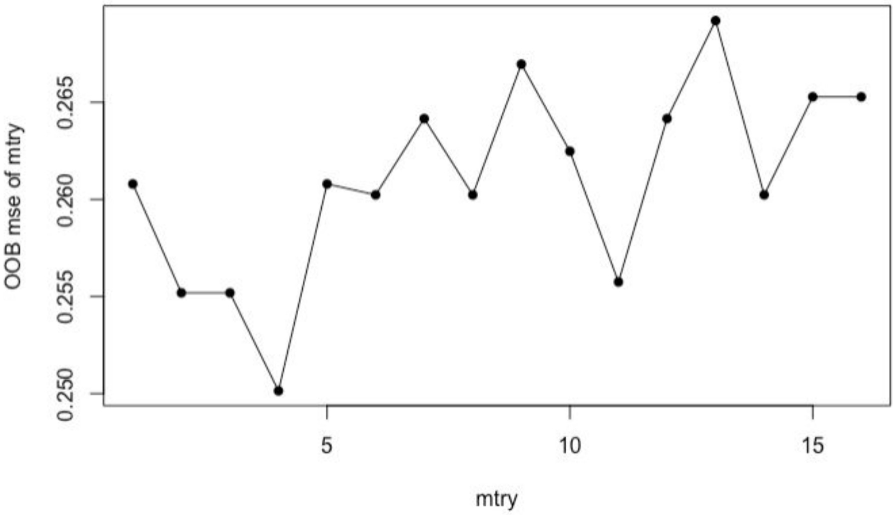
6. Tuning Random Forest

Ntrees



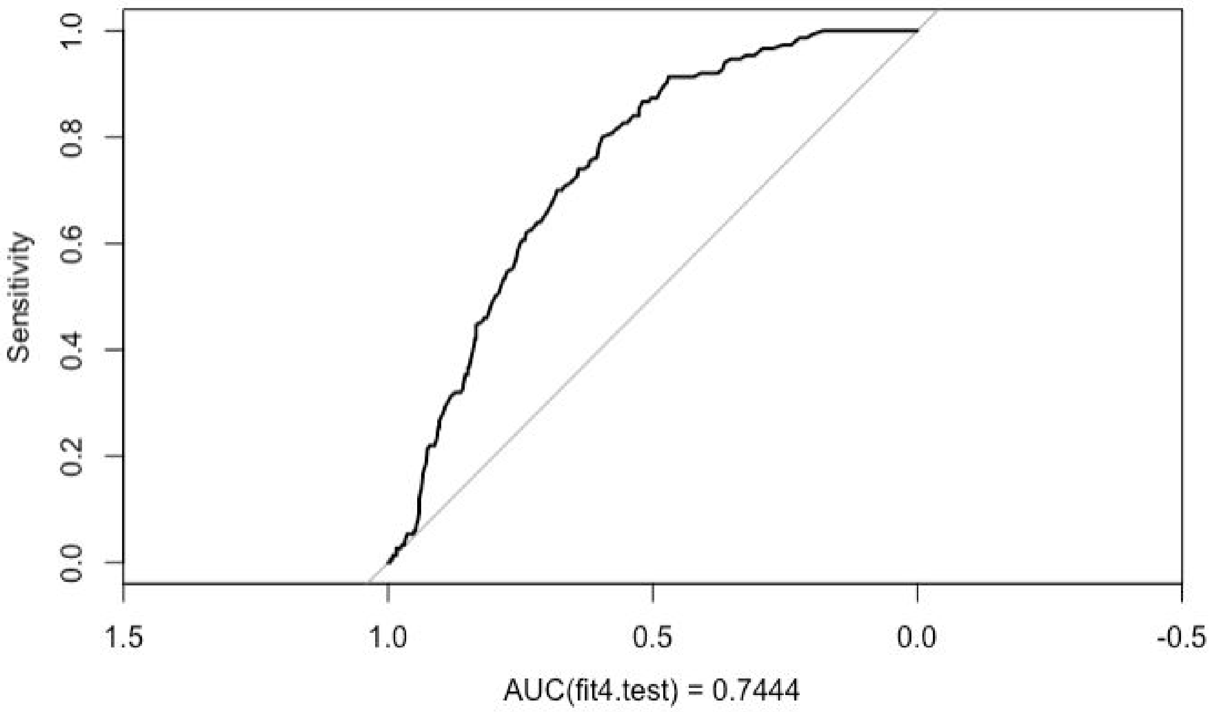
We chose to use **ntree = 250**.

Mtry



We choose **mtry = 4**.

7. Random Forest ROC Curve


Sensitivity
1.0
0.8
0.6
0.4
0.2
0.0
1.5
1.0
0.5
0.0
-0.5
AUC(fit4.test) = 0.7444