**Income Qualification**

**DESCRIPTION**

Identify the level of income qualification needed for the families in Latin America.

**Problem Statement Scenario:**

Many social programs have a hard time ensuring that the right people are given enough aid. It’s tricky when a program focuses on the poorest segment of the population. This segment of the population can’t provide the necessary income and expense records to prove that they qualify.

In Latin America, a popular method called Proxy Means Test (PMT) uses an algorithm to verify income qualification. With PMT, agencies use a model that considers a family’s observable household attributes like the material of their walls and ceiling or the assets found in their homes to classify them and predict their level of need.

While this is an improvement, accuracy remains a problem as the region’s population grows and poverty declines.

The Inter-American Development Bank (IDB)believes that new methods beyond traditional econometrics, based on a dataset of Costa Rican household characteristics, might help improve PMT’s performance.

**Analysis Tasks to be performed:**

1. Identify the output variable.
2. Understand the type of data.
3. Check if there are any biases in your dataset.
4. Check whether all members of the house have the same poverty level.
5. Check if there is a house without a family head.
6. Set poverty level of the members and the head of the house within a family.
7. Count how many null values are existing in columns.
8. Remove null value rows of the target variable.
9. Predict the accuracy using random forest classifier.
10. Check the accuracy using random forest with cross validation.

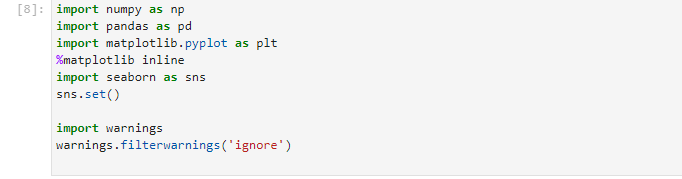
**Core Data fields**

1. Id - a unique identifier for each row.
2. Target - the target is an ordinal variable indicating groups of income levels.  
   1 = extreme poverty 2 = moderate poverty 3 = vulnerable households 4 = non vulnerable households
3. idhogar - this is a unique identifier for each household. This can be used to create household-wide features, etc. All rows in a given household will have a matching value for this identifier.
4. parentesco1 - indicates if this person is the head of the household.

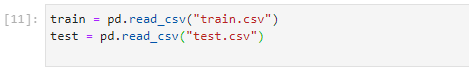
Note: that ONLY the heads of household are used in scoring. All household members are included in train +test data sets, but only heads of households are scored.

**Understand the Data**

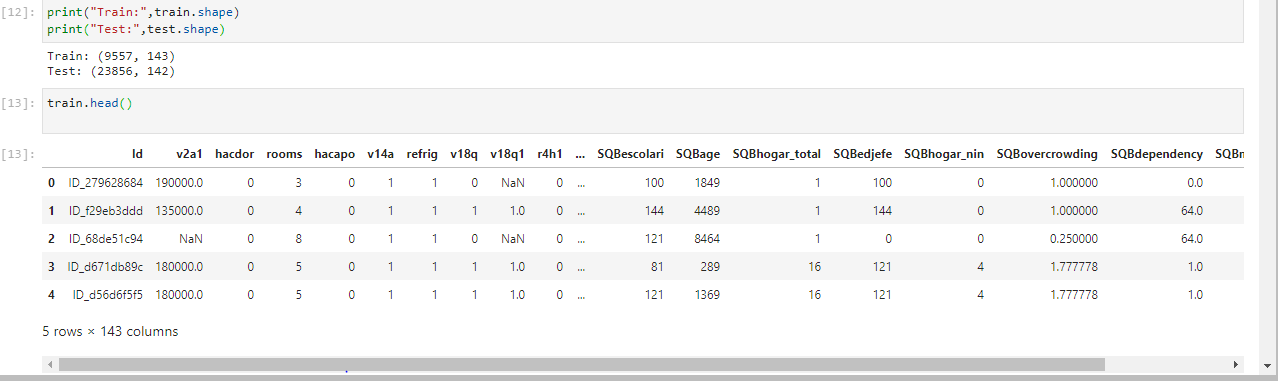
**Import necessary Libraries**

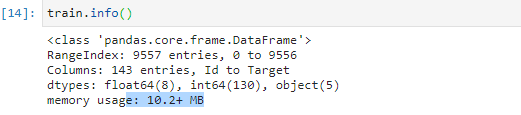
****

**Load Data**

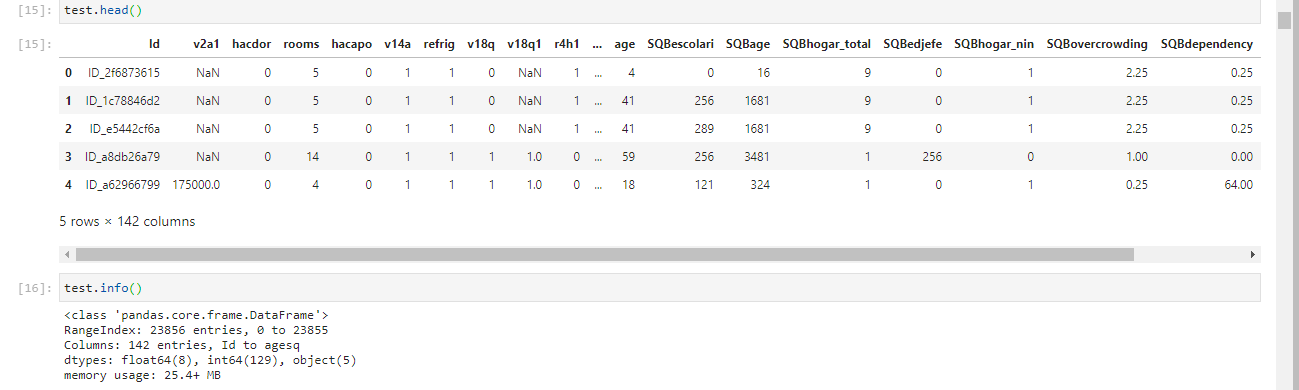
****

**Explore Train dataset**

****

****

**Explore Test dataset**

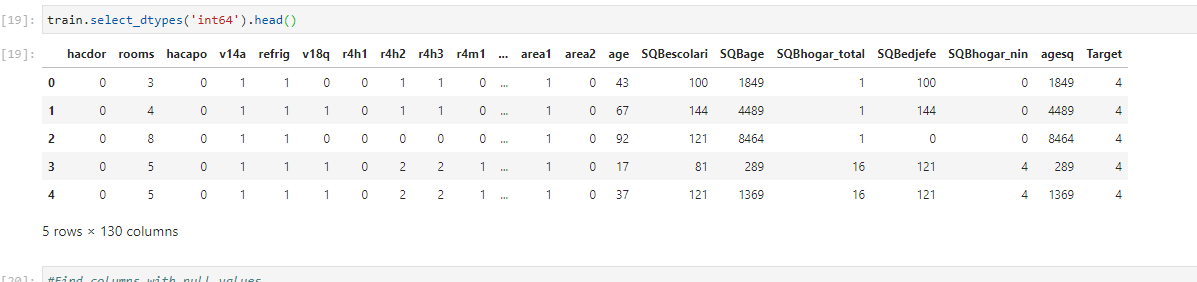
****

Looking at the train and test dataset we noticed that the following:  
Train dataset:  
Rows: 9557 entries, 0 to 9556  
Columns: 143 entries, Id to Target  
Column dtypes: float64(8), int64(130), object(5)

Test dataset:  
Rows: 23856 entries, 0 to 23855  
Columns: 142 entries, Id to agesq  
dtypes: float64(8), int64(129), object(5)

The important piece of information here is that we don’t have ‘Target’ feature in Test Dataset. There are 5 object type, 130(Train set)/ 129 (test set) integer type and 8 float type features. Lets look at those features next.





**Find columns with null values**

****

****

Looking at the different types of data and null values for each feature. We found the following:

1. No null values for Integer type features.

2. No null values for float type features.

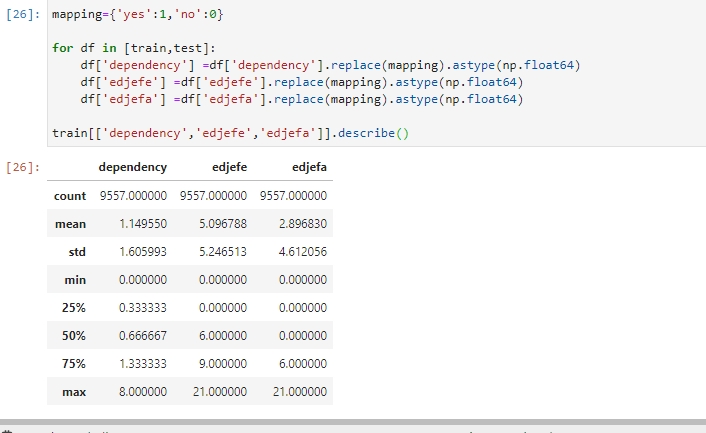
3. For Object types v2a1 6860 v18q1 7342 rez\_esc 7928 meaneduc 5 SQBmeaned 5

**Data Cleaning**

**Lets fix the column with mixed values.**

According to the documentation for these columns:  
dependency: Dependency rate, calculated = (number of members of the household younger than 19 or older than 64)/(number of member of household between 19 and 64)  
edjefe: years of education of male head of household, based on the interaction of escolari (years of education), head of household and gender, yes=1 and no=0  
edjefa: years of education of female head of household, based on the interaction of escolari (years of education), head of household and gender, yes=1 and no=0

For these three variables, it seems “yes” = 1 and “no” = 0. We can correct the variables using a mapping and convert to floats.



**Lets fix the column with null values**

According to the documentation for these columns:

v2a1 (total nulls: 6860) : Monthly rent payment  
v18q1 (total nulls: 7342) : number of tablets household owns  
rez\_esc (total nulls: 7928) : Years behind in school  
meaneduc (total nulls: 5) : average years of education for adults (18+)  
SQBmeaned (total nulls: 5) : square of the mean years of education of adults (>=18) in the household 142

