STAT - 515
Final Project

Costa Rican Household Poverty Level Prediction

Chaithanya Pramodh Kasula (G01197109) & Aishwarya Varala (G01206112)

Introduction: The current report details the process of answering several research questions related to the poverty levels of Costa Rican households. It comprises data sources, exploratory data analysis through visualization, model development, fine-tuning, approaches to tackle data imbalance problems, performance metrics and visualization of results.

Background: In a state or a locality, it is important for the government or banks to identify the right households which are in need of aid for their social welfare programs. It was observed that people living in the economically backward areas, do not have the necessary knowledge or cannot provide the necessary documents such as income/expense records to prove that they qualify for the aid.

In Latin America, a popular method, called the Proxy Means Test, is used to make that decision. Agencies look at a family's observable attributes such as the material of the ceiling, number of rooms in the household, number of people in the household, etc, to conclude about a family's qualification for the aid. However, accuracy remains a problem. Hence, the Inter-American Development Bank has provided a data set to the Kaggle community to come up with new methods that could effectively direct them towards the households that are in need of social welfare assistance.

Data Source: The data set used in the project is extracted from Kaggle. The URL of the data source is:
https://www.kaggle.com/c/costa-rican-household-poverty-prediction/overview

Data Description: The data folder consists of train.csv and test.csv with 9557 rows and 23856 rows respectively. However, the test.csv does not contain the 'target' column which determines the poverty level. Hence, train.csv alone is used as the data set whose size is 3.08 MB. The number of columns are 143. Each record is associated with a single person. The descriptions of 143 columns are found in the URL of the data source mentioned above. The descriptions for a few columns are provided below.

Target: denotes poverty level
1 = extreme poverty, 2 = moderate poverty, 3 = vulnerable households, 4 = non-vulnerable household

Idhogar: A unique identifier for each household. People belonging to a single household are identified by this column.

v2a1: Monthly rent paid by each household.

rooms: the number of all rooms in the house.

escolari: years of schooling etc.

Associated Research Questions:

R1: Can we construct a model to identify the level of poverty for various Costa Rican households?

R2: Can we identify the most important factors/columns/predictors that determine the level of poverty for a household?

R3: Is there any relationship between the education attained, gender, head of the household, number of persons, number of rooms in the household, dependency, and technology (mobile phones, computer, television, tablet) to the level of poverty for a household?

R4: In the absence of the 'target' column, and with the given features in R3, how accurately can K-Means clustering algorithm help in assigning the class label (determining the poverty level/values of the target column) for a person?

Data types of variables: There are four data types associated with the variables in the data set:

1. Boolean: Integer Boolean (0 or 1), Character Boolean (yes or no). Columns such as paredblolad, noelec, etc.
2. Float data type. For example, meaneduc, overcrowding, etc.
3. Integer data type. For example, age, rent, rooms, tamviv, etc.
4. Alpha-numeric. For example, Id, idhogar.

Data Exploration

Class distribution: The poverty level distribution (class distribution) is very imbalanced as shown in Fig. 2. The number rows belonging to class 'four' form 65.72% of the data set but the number of rows belonging to class 'one' account to only 0.074%. The same uneven distribution has been observed in the household-level data set as depicted in Fig. 3. For a detailed understanding of household-level data set please read the **'Household data set'** section.

Number of NaN values in monthly_rent_payment column: There are 6860 rows that contain NaN values in the 'monthly_rent_payment' column. The columns, 'own_and_fully_paid_house', 'own_paying_in_installments', 'rented', 'precarious' and 'other_assigned_borrowed' contain binary values that denote 0 (FALSE) or 1 (TRUE). From Fig. 4, it can be inferred that there are 5911 people who own a house and 961 people who own the house but pay installments. This fact can be immensely useful during data pre-processing.

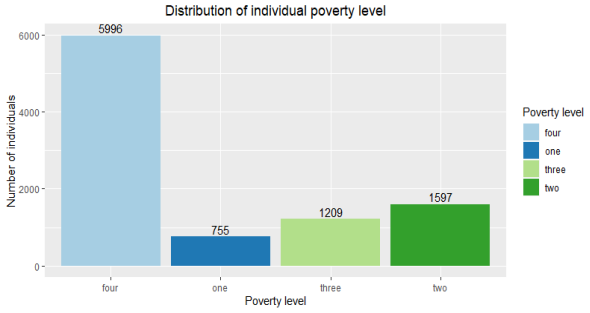


Fig. 2

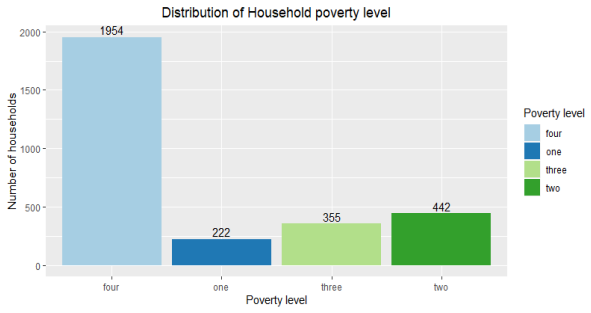


Fig. 3

Data pre-processing:

The original column names have been renamed to their shortened English equivalent descriptions for easy reference, understanding, and communication. They can be found in the 'changed_column_names.csv'. Henceforth, data will be referenced through the renamed columns.

Missing value treatment and Feature Engineering: From the research questions, R1, R2, and R3, it can be interpreted that the unit of analysis is a household. However, each record in the data set describes the attributes of a single person in the household. People belonging to a single household can be grouped together by the 'household_identifier' column as they have the same identifier value assigned to each of them. 'Household_identifer' values are unique to every household. Additionally, the same value of the 'target' class (poverty level) is assigned to all the people in a single household.

Missing values are found across multiple columns in the data set. With the explanations drawn in the **'Data Exploration'** section over the NaN values present in the 'monthly_rent_payment' column, it is assumed that all the people who own a house do not pay the rent. There are only 7 people left. Since their count is too less, all the rows with NaN values in the 'monthly_rent_payment' column are replaced with a zero.

The column 'number_of_tablets_household_owns' also contains NaN values. The column 'owns_a_tablet' indicates whether a household owns a tablet or not. If the household does not own a tablet, then its value in the 'number_of_tablets_household_owns' column is replaced with a zero. For every household, the mean values for the columns 'years_of_schooling' and 'years_behind_in_school' are computed and assigned to the household head. The character Boolean values (yes or no) in the 'dependency' column have been replaced with 1 or 0 respectively. The same operation has been performed for 'edjefe' and 'edjefa' columns.

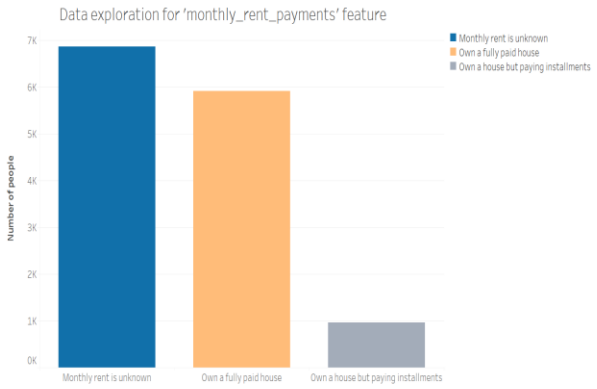
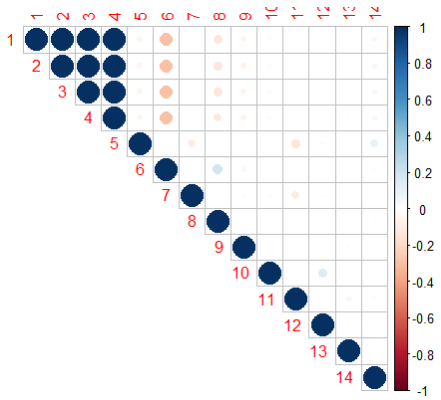


Fig. 4



Correlation plot between highly correlated features

Fig. 5

Duplicate columns are removed from the data set. For example, there are two columns with the same name 'age_squared' that are duplicates of each other. Only, one of them is retained. Additionally, there are a lot of columns that are squared values of existing columns such as overcrowding_squared, dependency_squared, etc. All such columns have been removed from the data set as they provide no additional information to the model. Also, the class variable, 'target', consists of poverty levels in numeric format 1, 2, 3 and 4. They have been replaced with words one, two, three and four respectively. The columns (if_stepson_or_doughter, if_son_or_doughter_in_law, if_grandson_or_doughter, if_father_or_mother_in_law, if_brother_or_sister,if_brother_or_sister_in_law, if_other_family_member, if_spouse_or_partner, if_son_or_doughter, if_mother_or_father) are unimportant and do not fall under the scope of answering the research questions and therefore have been eliminated. The packages 'dplyr' (Wickham et al., 2017), and 'stringr' (Wickham, 2019) are used for data pre-processing. The packages 'ggplot' (Wickham, 2016) and Tableau (Tableau Software, 2019) software were utilized for data visualization. The package 'corrplot' (Wei et al., 2017) was used for plotting the correlation matrix.

Using Correlation to reduce features: A correlation matrix has been constructed for 119 columns which remain after the pre-processing stage. Visualization of such a huge plot is clumsy. Hence, highly correlated features whose correlation value is greater than 0.98 have been extracted from the matrix and plotted separately. As the software cannot plot cannot incorporate lengthy column names, they have been represented with numbers in Fig. 5. The numbers 1 to 14 in the picture correspond to the columns 'size_of_the_household', 'household_size' , 'of_total_individuals_in_the_household', 'total_persons_in_the_household', , 'toilet_connected_to_sewer_or_cesspool', 'if_household_head', 'region_brunca', 'if_widower', 'no_main_source_of_energy_used_for_cooking_no_kitchen','if_predominant_material_on_the_outside_wall_is_natural_fibers', 'electricity_from_cooperative', 'if_predominant_material_on_the_roof_is_natural_fibers', 'if_predominant_material_on_the_floor_is_wood' and 'if_predominant_material_on_the_roof_is_fiber_cement_mezzanine' respectively. The purpose of constructing a correlation plot is to remove highly correlated columns from the data set as they do not provide any additional value. From Fig. 5, it is observed that 'size_of_the_household', 'household_size', 'of_total_individuals_in_the_household', 'total_persons_in_the_household' are highly correlated to each other. Hence, only one of those columns is included in the data set.

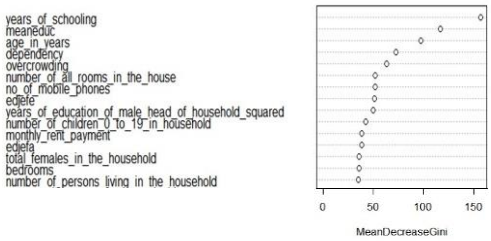
Household data set: The household head is regarded as a representative of each household. Hence, only rows whose 'if_household_head' column equals 1 are made a part of this data set. Features such as 'years_of_schooling' that are associated with a single person have been appropriately handled to reflect the household during data pre-processing. A total of 2973 rows form this data set and henceforth, it will be referenced as the household dataset.

Modelling using Random Forest: In order to answer the first research question, the household data set is used to train a random forest. For a given instance, whose poverty level/target is unknown, the trained model would predict the class for that row. Random Forest is an ensemble learning technique that builds trees of varying lengths by taking samples from the data set (bootstrap sampling). The left-over data which is not a part of the construction is often called

the Out-Of-Bag (OOB) data set. The constructed model then uses the OOB data set as a test set and evaluates its performance on its own. Assuming the rows are independent of each other (as in our case), there is no need to perform cross-validation separately while using Random Forest. It is implicitly done internally with the help of OOB data. The OOB error for each constructed decision tree can be averaged to represent the model's overall error rate. Random Forests also generalize well and avoid overfitting which is one of the major problems observed in Decision Trees. A package called 'randomForest' (Liaw et al., 2002) from the 'caret' package (Kuhn et al., 2019) is used for training and testing the data.

Splitting the data set into Train and Test sets: Due to the class imbalance problem, randomly splitting the data set in the ratio of 75:25 for training and test sets does not extract a significant number of rows associated with the minority class in the test set. Hence, the obtained performance metrics are not very reliable. Therefore, 75% of data from each class is made a part of the training set and 25% of data from each class was made a part of the test set. Therefore, 75% and 25% of data from each class constitute the train and test set respectively. The resultant number of rows in the train and test set are 2230 and 744 respectively.

Training: For the first iteration, 112 columns (columns left after pre-processing) were used for training. Alphanumeric features such as 'id' and 'household_identifier' columns were removed from the training set. The hyperparameters used for training the classifier are: 'ntree=500' and 'mtry=10'. Various 'mtry' values were tested but 'mtry' equalling 10 resulted in better performance. In order to reduce the number of columns used for training, the MeanDecreaseinGini values are extracted from the trained model. The value of the mean decrease in Gini is directly associated with the importance of a feature. The greater the mean decrease in Gini value for a feature, the more its importance in predicting the target variable. Fig. 6 represents the top 15 important features. In the coming iterations, only these 15 features are used for training. This reduces the cost of training by decreasing the number of features from 112 to 15.



Top fifteen important features from the random forest model

Fig. 6

```
        OOB estimate of  error rate: 31.43%
Confusion matrix:
      four one three two class.error
four  1409   8     6  43  0.03888131
one    104  19     2  41  0.88554217
three  204   4    14  44  0.94736842
two    216  14    15  87  0.73795181
```

OOB estimate and Confusion Matrix without Sampling

Fig. 7

The OOB error estimate is shown in Fig. 7. It can be noticed that the class error for 'four' is very less. Whereas, the class error for 'one', 'two' and 'three' are high. The model was trained well on class 4 because the number of records for class 4 are relatively high and hence low error. But, due to the lesser number of records associated with 'one', 'two' and 'three' classes, the model was not trained well, hence the high error.

Sampling: In order to adjust the distribution of classes in the data set, two popular techniques known as under-sampling and oversampling have been employed. Under-sampling a class involves taking only a fraction of records associated with the majority class. To illustrate, only a few records from class 'four' are extracted and made a part of the data set. The records belonging to other minority classes are unchanged. Under-sampling results in loss of data. Oversampling involves synthesizing/duplicating records belonging to the minority classes.

Under-sampling: Random under-sampling has been performed for class 'four'. 35% of the records belonging to class 'four' have been chosen randomly and made part of the data set. So, the number of records belonging to class 'four' was reduced from 1954 to 684. No sampling has been performed on the records belonging to other classes. After under-sampling, Fig. 8 and Fig. 9 show the class distribution for train and test sets respectively. Fig. 10 details the OOB error estimate for the under-sampled data set. It can be observed that the OOB error estimate for classes 'one', 'two' and 'three' has not reduced significantly when compared to Fig. 7. The class error for 'four' has increased. The model does not perform well over any class due to the lesser number of records available for all the classes. Hence,

under-sampling the data set when a lesser number of records are available is a bad approach as it can lead to underfitting.

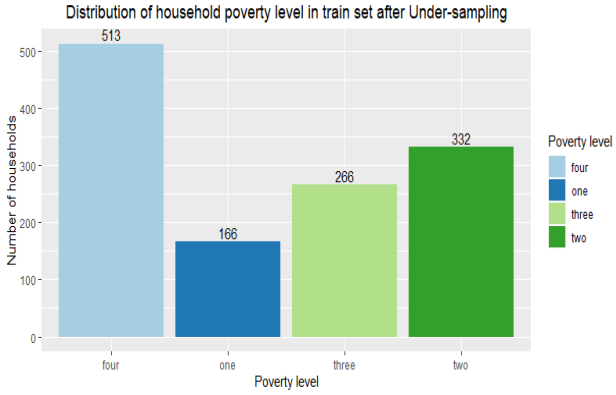


Fig. 8

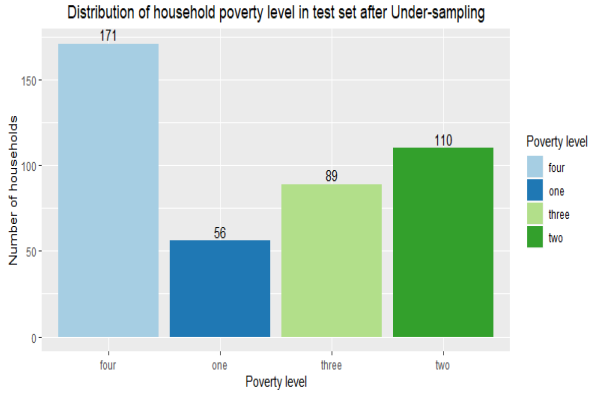


Fig. 9

```
        OOB estimate of  error rate: 50.74%
Confusion matrix:
      four one three two class.error
four   395   7    43  68   0.2300195
one     42  28    17  79   0.8313253
three  102   7    60  97   0.7744361
two    111  23    52 146   0.5602410
```

OOB estimate and Confusion Matrix during Under-sampling

Fig. 10

```
        OOB estimate of  error rate: 12.02%
Confusion matrix:
      four one three two class.error
four  1330  26    33  77  0.09276944
one     15 479     1   5  0.04200000
three   84   5   413  30  0.22368421
two     66  18    20 559  0.15686275
```

OOB estimate and Confusion Matrix during Oversampling

Fig. 11

Oversampling: Random oversampling has been performed for classes ‘one’, ‘two’ and ‘three’. The records associated with the mentioned classes have been duplicated appropriately to minimize the difference in class distribution. After oversampling, Fig. 12 and Fig. 13 represent the class distribution in train and test sets respectively. The model has been retrained and the OOB error estimate is represented in Fig. 11.

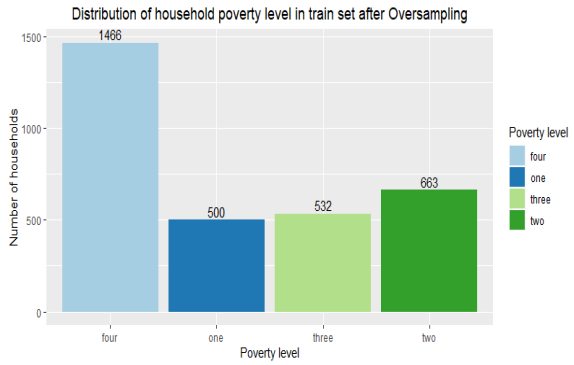


Fig. 12

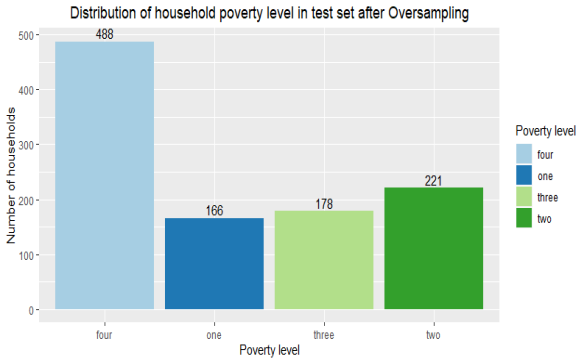


Fig. 13

It can be noticed that there is a significant decrease in the over-all OOB error estimate and the respective class errors. Hence, oversampling the records of the minority classes promotes better model training and development. Fig. 14 and Fig. 15 represent the change in the training error rate of different classes with the increase in the number of trees during under-sampling and oversampling. The colors red, green, blue and purple represent classes ‘four’, ‘one’, ‘two’ and ‘three’ respectively. The black line represents the over-all OOB error rate. During oversampling, the error rate for all the classes decreases with the increase in the number of trees.

Testing: The total number of records in the test set is 488. It comprises 25% of records extracted from each class. The columns ‘target’, ‘id’ and ‘household_identifier’ have been removed from the test set. The resultant data is sent to the trained random forest classifier to obtain results. For each record in the test set, the classifier utilizes the knowledge

acquired through training to predict the class for that record. The predicted labels are evaluated against their original values in the 'target' class which determines the model's performance.

Performance Metrics: Accuracy is one of the primary metrics for evaluating classification models. It is defined as follows:

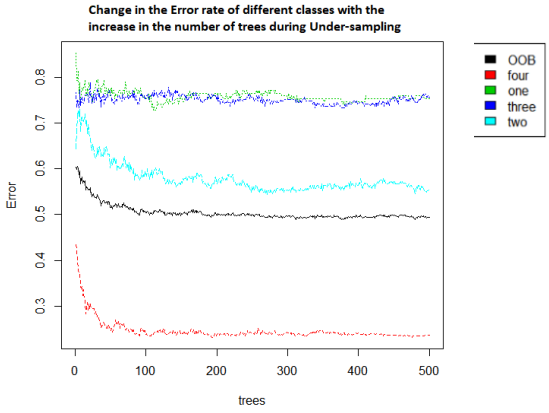


Fig. 14

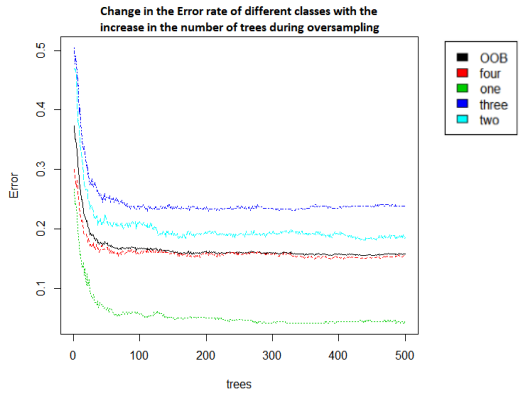


Fig. 15

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ number\ of\ Predictions}$$

However, in classification problems, Accuracy alone cannot be used to evaluate a classifier. A closer inspection of the obtained results can be made through a Confusion Matrix. A confusion matrix is a combination of predicted and actual values of different classes. Important metrics such as Precision (Specificity) and Sensitivity (Recall) can be derived from it. Sensitivity is also called as true positive rate. Specificity is also known as the true negative rate. The higher the values of Accuracy, Sensitivity, and Specificity for the existing classes, the better the model.

$$Sensitivity = \frac{Number\ of\ True\ Positives}{Number\ of\ True\ Positives + Number\ of\ False\ Negatives}$$

$$Specificity = \frac{Number\ of\ True\ Negatives}{Number\ of\ True\ Negatives + Number\ of\ False\ Positives}$$

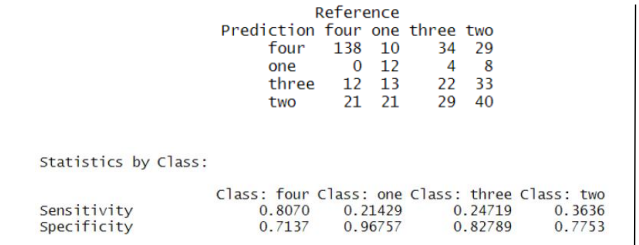
The function 'confusionMatrix' from the caret (Kuhn et al., 2019) package is used to obtain the confusion matrix by providing actual and predicted values as input. Fig. 16 and Fig 17 represent the confusion matrices and the classifier's performance for the respective techniques. However, the metrics represented in the population must be adjusted as these metrics are for the extracted samples but not for the whole population. Therefore, the obtained metrics must be adjusted so that they reflect the actual population.

For a sample, if C1, C2, C3, and C4 denote a classification metric for classes 'one', 'two', 'three' and 'four' respectively, then, its weighted metric value for the whole/original population is denoted by,

$$Weighted\ Classification\ Metric = \frac{\frac{C1N1}{S1} + \frac{C2N2}{S2} + \frac{C3N3}{S3} + \frac{C4N4}{S4}}{\frac{N1}{S1} + \frac{N2}{S2} + \frac{N3}{S3} + \frac{N4}{S4}}$$

where N1, N2, N3, N4 represent the original/actual population size and S1, S2, S3, S4 represent the sample population size. Therefore the weighted Accuracy, Weighted Specificity, Weighted Sensitivity for the classifier trained on under-sampled data are 53.41%, 78.65%, and 18.01% respectively. The Weighted Accuracy, Weighted Specificity, Weighted Recall for classifier trained on oversampled data set are 84.39%, 93.24%, and 84.44%, respectively.

Evaluating the relationship between the features mentioned in R3 and the poverty level: In order to evaluate the relationship between education attained, gender, head of the household, number of persons, number of rooms in the household, dependency, and technology (mobile phones, computer, television, tablet) to the level of poverty, the model is trained with only the below mentioned features. The trained model is then used to predict the records in the test set. The closer the performance metrics of the new classifier, to the performance metrics of the old classifier (model trained with the top 15 important features), the stronger the relationship between the mentioned features and the 'target' (poverty level).



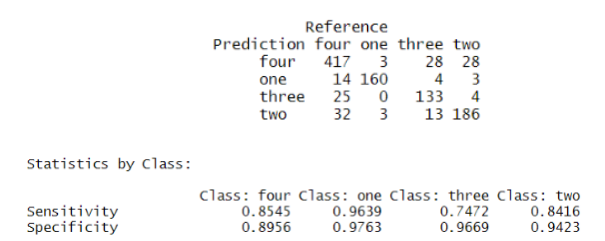
Reference				
Prediction	four	one	three	two
four	138	10	34	29
one	0	12	4	8
three	12	13	22	33
two	21	21	29	40

Statistics by Class:

	Class: four	Class: one	Class: three	Class: two
Sensitivity	0.8070	0.21429	0.24719	0.3636
Specificity	0.7137	0.96757	0.82789	0.7753

Performance Metrics of the classifier on test set during Under-sampling

Fig. 16



Reference				
Prediction	four	one	three	two
four	417	3	28	28
one	14	160	4	3
three	25	0	133	4
two	32	3	13	186

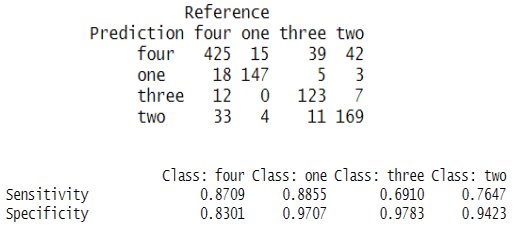
Statistics by Class:

	Class: four	Class: one	Class: three	Class: two
Sensitivity	0.8545	0.9639	0.7472	0.8416
Specificity	0.8956	0.9763	0.9669	0.9423

Performance Metrics of the classifier on test set during Oversampling

Fig. 17

The new model is trained by oversampling classes 'one', 'two' and 'three' but only using the features 'edjefa', 'edjefe', 'years_of_education_of_male_head_of_household_squared','dependency', 'overcrowding', 'meaneduc', 'years_of_schooling', 'total_females_in_the_household', 'total_persons_in_the_household', 'no_of_mobile_phones', 'total_males_in_the_household', 'if_the_household_has_notebook_or_desktop_computer'. Fig. 18 details the obtained performance metrics of the retrained model over the test set.



Reference				
Prediction	four	one	three	two
four	425	15	39	42
one	18	147	5	3
three	12	0	123	7
two	33	4	11	169

	Class: four	Class: one	Class: three	Class: two
Sensitivity	0.8709	0.8855	0.6910	0.7647
Specificity	0.8301	0.9707	0.9783	0.9423

Performance Metrics of the classifier on the test set by the trained model with features mentioned in R3

Fig. 18

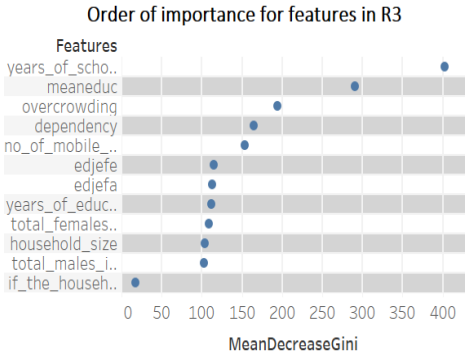
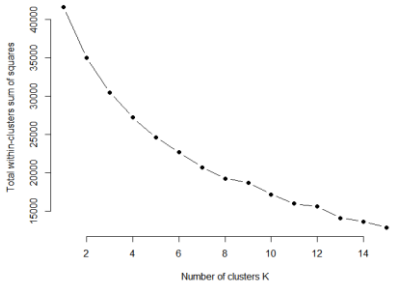


Fig. 19

The over-all adjusted Weighted Accuracy, Weighted Sensitivity and Weighted Specificity for the original population are 81.11%, 0.8115 (81.15%), and 0.9059 (90.59%), respectively. The results show that the features exhibit a strong association with the 'target' class. Their sorted MeanDecreaseGini values as shown in Fig. 19. Among the selected features, 'years_of_schooling' is the most important feature and 'if_the_household_has_notebook_or_desktop_computer' is the least important.

Clustering: Clustering techniques facilitate the grouping of records based on a similarity measure. K-Means is a famous clustering technique which uses a distance metric (generally Euclidean) to cluster the data points. The centroid is the most representative point of a K-Means cluster. The packages 'cluster' (Rousseeuw et al., 2019), 'factoextra' (Kassambara et al., 2017) and 'purrr' (Henry et al., 2019) are used for the K-Means algorithm, cluster visualization and elbow plot (Fig. 20) respectively. The features that were used in '**Evaluating the relationship between the features mentioned in R3 and the poverty level**' section were only used for clustering as they are related to the factors mentioned in R3. K-Means algorithm was applied over the mentioned features only. Here, the unit of analysis is a single person. The input cluster number was provided as 8, as it appears to be the bend on the knee in the elbow

graph (Fig. 20). The pre-processed data that was used for random forest is also utilized for K-Means. The 'target' column is removed from the data set.



Elbow Plot

Fig. 20

Visualization of K-Means Clusters

Fig. 21

Fig. 21 shows the visualization of 8 clusters. The X and Y axes represent the reduced dimensions of records that are represented by two principal components with the highest variance. The dimensionality reduction was performed with PCA through the 'factoextra' package for visualization only. In K-Means, each record is assigned to its nearest centroid which is represented by a cluster number. Each color in the plot represents a cluster.

A record and its cluster number are compared for further analysis. The purpose is to find whether K-Means has the ability to cluster records with similar poverty levels together. Table. 1 represents the distribution of individual poverty levels in the resultant clusters. It can be inferred that K-Means clustering is not effective in determining the poverty level as there was no association between the resultant clusters and the original poverty levels. The reason can be attributed to the unequal class distributions and non-linearity of data. While minimizing the within-cluster sum of squares, the algorithm gives more weight to large clusters than the smaller ones. Hence, no clear groups were observed. The poverty levels were distributed across clusters. In order to check the importance of the input cluster number, different values were provided as input and K-Means clusters were generated accordingly. However, an increase in the cluster number did not result in the formation of efficient clusters. As an experiment, the top 15 features (which are all continuous variables), obtained from the trained random forest model were used instead of the features in R3, to generate the K-Means clusters. This also did not improve the result significantly.

	Four	One	Two	Three
1	429	40	71	78
2	682	58	111	117
3	17	17	11	18
4	141	12	35	38
5	59	18	14	27
6	5	4	2	6
7	133	50	45	78
8	488	23	66	80

Distribution of individual poverty levels in the clusters (when input number of clusters = 8)

Table. 1

Answers to Research Questions:

Answer for R1: Yes. The Random Forest classifier has been constructed to successfully identify the level of poverty for Costa Rican households with good performance metrics. With the availability of more data, the performance of the model can be improved.

Answer for R2: Yes. The fifteen most important columns that determine the level of poverty for a household, in the decreasing order of their importance are, 'years_of_schooling', 'meaneduc', 'age_in_years', 'dependency', 'overcrowding', 'number_of_all_rooms_in_the_house', 'no_of_mobile_phones', 'edjefe', 'years_of_education_of_male_head_of_household_squared', 'number_of_children_0_to_19_in_household',

'monthly_rent_payment', 'edjefa', 'total_females_in_the_household', 'bedrooms', 'number_of_persons_living_in_the_household'.

Answer for R3: Yes. There is a strong relationship between the features related to the entities mentioned in the question R3 to the level of poverty for a household.

Answer for R4: K-Means clustering technique does not perform well in clustering the records with the same poverty level together. Hence, in the absence of the 'target' column, and with the given features in R3, K-Means clustering algorithm cannot help in assigning the class label for a person/individual.

Tasks performed:

Chaithanya Pramodh Kasula	Aishwarya Varala	Collective efforts
Data pre-processing	Data Visualization	Data Exploration
Feature Engineering	K-Means Clustering	Result Analysis
Random Forest Model Construction	Correlation plots	Building Research Questions
		Report Writing

Challenges:

1. The chosen dataset had 143 columns which made data preprocessing and feature engineering a challenge.
2. Transforming data from an individual level to household level was a laborious task.
3. Application of sampling and adjusting the obtained performance metrics to reflect the original population.
4. Faced problems with the 'corrplot' package for displaying the plot due to lengthy column names.
5. Plotting the K-Means clusters and experimenting with various inputs of K-Means clustering.
6. It was difficult to assign colorblind-friendly, print-friendly and photocopy safe colors. But, they were used as a part of visualization wherever possible.

References:

Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2019). dplyr: A Grammar of Data Manipulation. R package version 0.8.3. https://CRAN.R-project.org/package=dplyr

Hadley Wickham (2019). stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.4.0. https://CRAN.R-project.org/package=stringr

H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.

Tableau Software (2019). Retrieved from https://www.tableau.com/

Taiyun Wei and Viliam Simko (2017). R package "corrplot": Visualization of a Correlation Matrix (Version 0. 84). Available from https://github.com/taiyun/corrplot

A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. R News 2(3), 18--22.

Max Kuhn. Contributions from Jed Wing, Steve Weston, Andre Williams, Chris Keefer, Allan Engelhardt, Tony Cooper, Zachary Mayer, Brenton Kenkel, the R Core Team, Michael Benesty, Reynald Lescarbeau, Andrew Ziem, Luca Scrucca, Yuan Tang, Can Candan and Tyler Hunt. (2019. caret: Classification and Regression Training. R package version 6.0-84. https://CRAN.R-project.org/package=caret

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., Hornik, K.(2019). cluster: Cluster Analysis Basics and Extensions. R package version 2.1.0.

Alboukadel Kassambara and Fabian Mundt (2017). factoextra: Extract and Visualize the Results of Multivariate Data Analyses. R package version 1.0.5. https://CRAN.R-project.org/package=factoextra

Lionel Henry and Hadley Wickham (2019). purrr: Functional Programming Tools. R package version 0.3.2. https://CRAN.R-project.org/package=purrr