Machine Learning and Data Science for a Household-Specific Poverty Level Prediction Task

By

Sudesh Kumar Venkatramolla

B.E., Chaitanya Bharathi Institute of Technology, Telangana, India 2017

A REPORT

Submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computer Science
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2019

Approved by:

Major Professor
Dr. William H Hsu

Copyright

Abstract

This project focuses on a prediction task from the Kaggle data science challenge site: prediction of the poverty level of individual households using supervised classification learning. In Latin America, the Proxy Means Test (PMT) is the most popular method used to verify the income qualification. The PMT works by considering the observable properties of a household, such as the walls, ceilings, and electric devices in a family home. These and other general assets are used to classify the poverty level, assigning one of the four labels: (1) extreme poverty, (2) moderate poverty, (3) vulnerable households and (4) non-vulnerable households. The accuracy of learned classification models submitted as solutions to this data challenge has tended to decrease as a function of dataset size. Therefore, in this project, I am focusing on methods for boosting accuracy in detecting poverty level using committee machines (bagging, boosting, etc.) for supervised inductive learning. Because the task is classification learning, my first approach is to apply random forests (a decision tree ensemble method); depending on the accuracy, I will proceed with the advanced methods, such as light gradient-boosting methods (GBMs) and neural networks that are frequently used on large, complex multivariate classification tasks. The inference task is to predict the poverty level of a new household using attributes of the family home and other attributes found to be relevant by the learning algorithm. This enables use of cases of artificial intelligence for social good, such as helping governments and relief and economic development agencies to identify communities in need.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my committee members Dr. William H. Hsu, Dr. Mitchell Neilsen, and Dr. Torben Amtoft for taking time to serve on my committee and their support in the process of this project.

I am especially indebted to my major advisor Dr. William H. Hsu for believing in my abilities and for his constant support from my very first semester here at Kansas State University.

I am also grateful to Dr. Torben Amtoft, Dr. Daniel Anderson, and Dr. John Keller for being amazing mentors during my Teaching Assistant phases.

I would like to extend a huge thanks to all my family members without whom this journey would not have been possible. I am extremely grateful to my mom, Anuradha Venkatramolla, and my dad, Sudhakar Venkatramolla for their unconditional love, encouragement, and emotional support in my life. I cannot thank my sister, Srini enough for her immense love and support. She is my strength and has played a significant role in what I have achieved so far in my life.

And last, but in no way the least, I would like to thank my friends and seniors Pavan Manepalli, Teja Usha Sree Vallabhaneni, Rahul Chandra Reddy, Sindhu Velumula, Sneha Gullapalli, Yojitha Reddy, Poojitha Bikki, Pruthvidhar Dhodda, Nithin for their support and friendship. The past two years would not have been so memorable and fun without all of them.

Table of Contents

List of Figures

List of Tables

Chapter 1- Introduction

This chapter presents a brief synopsis of this project, starting with the problem statement, overall goal, and objectives. It gives an overview of relevant extant methods for this data challenge and a rationale for the experimental approach taken and the evaluation methods used.

1.1 Problem Definition

The elimination of poverty worldwide is the first of 17 UN Sustainable Development Goals for the year 2030 (Pandey, 2018). According to the annual report on Costa Rica's "State of Nation", from 2017, 20 percent of households were in a situation of poverty and exclusion. Despite some improvements, such as the fall of the percentage of household poverty situations between 2015 and 2016, last year 31.5 percent of Costa Rican households suffered from poverty - monetary, multidimensional, and other types. Also, "The State of the Nation Report 2017" states that Costa Rica has failed to name some of the structural problems underlying poverty. These facts lead to a necessary action from Costa Rican authorities to fight these structural problems. Measuring poverty is currently notoriously difficult, time-consuming and expensive. Usually, estimates are done by collecting complex household consumption surveys with data consisting of several hundred different variables; each can be useful when accessing different poverty levels.

Machine learning offers new approaches for determining which variables are most productive. These algorithms can observe the poverty trend and the most important features when data of a certain period is examined. The main task of this work is to use machine learning algorithms to determine which households need the most financial support from the government agencies to improve their lifestyle.

1.2 Goals and Technical Objectives

The aim of this work is to build supervised inductive learning models which adopt classification methods to predict the poverty level of a household. In this work, I have tried to predict poverty using the several features of the household. For this work, I have used the dataset "Household Poverty of Costa Rica" from Kaggle. The inference task is to predict the poverty level of a new household using attributes of the family home and other attributes found to be relevant by the learning algorithm. I tried to classify poverty level into one of the following labels (1) extreme poverty, (2) moderate poverty, (3) vulnerable households and (4) non-vulnerable households.

1.3 Synopsis

The report comprises a brief review of the supporting literature for the task of predicting the poverty level of household and describes a machine learning and data science project centered on further explorations of a previously-developed experimental test bed. These involve extraction of data, cleaning the data, building the testing and training datasets for the supervised learning algorithms, training, and evaluation of models, and review of the models to derive actionable insights.

Various features in the dataset like mean education of the family, monthly rent of the house, wall quality, etc. are considered and feature selection was carried out to select the features that play the major role in predicting the poverty level of the household. The data is partitioned into testing and training datasets. Logistic regression, a supervised inductive learning algorithm, is used to train a classification model, which in turn is used to predict the poverty level of the household. Other classification algorithms such as Random forest and LightGBM are also be used to predict the poverty level of the household. These algorithms are used in feature selection and in the creation of a flexible model when data consists of a large set of features. The results of the algorithms were compared using the performance metrics such as precision, accuracy, and recall of these algorithms.

Chapter 2 -Background and Related Work

This chapter introduces the classification techniques used in this project. I have used the Logistic Regression, Random Forest and LightGBM algorithms and their associated discriminative representations to learn the classifier. The representation of classification algorithms and their optimization techniques are described.

2.1 Literature Survey: Classification Problem

Machine learning is training your machine to learn and solve problems without programming explicitly. Unsupervised learning deals with the unlabeled data and supervised learning is a machine learning algorithm technique which uses the labeled data to learn the mapping function between the input variable and the output variable. In supervised, we have the knowledge of the output variable. In supervised learning, it should learn a function from the given input and desired output values which could correlate them and derive a relationship from observable data. The supervised learning algorithms are classified into 2 types.

They are:

1. Classification Problems
2. Regression Problems

Classification:

Classification is a supervised learning technique which helps in predicting the output when they are categorical like the output is yes/No type or True/False type or when the people are categorized into young/Kid/Old age groups. Here the outputs are categorical, and Classification is mainly applied to the datasets that expect this kind of output.

2.2 Classification methods used in this work

2.2.1 Logistic Regression

The logistic regression is a predictive analysis that is like regression analyses. Logistic regression assumes that the dependent variable is binary and there are no outliers in data. It also assumes that there are no high correlations among the features that are using in predicting the dependent variable. This mainly concentrates on the task of estimating the log odds of an event. The logistic regression predicts the relation between the features and the dependent variable by estimating the probabilities using a logistic function. The function used in Logistic regression is the sigmoid function which intakes a real input and helps in outputting a value in the scale of 0 to 1 (Korkmaz, GÜNEY, & Ş, 2012).

2.2.2. Random Forest

Random forest is one of the most used algorithms that can be both used for classification and regression. It is one of the supervised learning algorithms. It builds a forest which is an ensemble of decision trees. Bagging method is used in this technique which is a combination of all learning models that increases the overall result. In general, Random forest will construct multiple decision trees and merges them to get a more accurate and stable prediction. Random Forest can easily measure the relative importance of various features to predict the dependent variable. This algorithm generally calculates the training and testing scores for each feature, so that the sum of all importance is equal to 1. Random forest prevents the problem of overfitting most of the time; it creates random subsets of features and builds smaller trees using these subsets (Breiman, 2001).

2.2.2. Light GBM

LightGBM is a gradient boosting framework that uses an algorithm that uses tree-based learning. It helps in fast training speed and achieving high efficiency. It has better accuracy and lower memory usage. LightGBM is capable of handling huge datasets and supports parallel and GPU learning. The LightGBM grows tree vertically while the remaining algorithms grow horizontally which indicates that grows the tree leaf-wise remaining algorithms grow the tree level-wise.

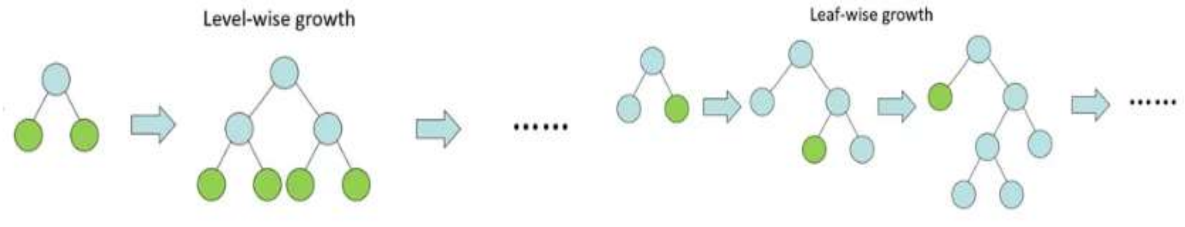

Fig 2.1: describes the difference between level-wise and leaf-wise growth

LightGBM is the most widely used algorithm in deep learning because

1. It is an open source and easy to use implementation
2. It is precise and fast

LightGBM uses a leaf-wise growth strategy while growing the tree. The difference between level-wise and leaf-wise approach is level-wise always tries to maintain the balance of the tree, but in the case of leaf-wise, it splits the tree to reduce the log-loss as minimum as possible. Level-wise training is prone to overfitting, but it is more flexible. Overfitting can be avoided here by using early stopping, finding the best split is the most important thing in Gradient Boosting Training algorithms, if it is done irregularly the complexity of the algorithm will be very high (Ke, Meng, Finley, & Wang, 2017).

There are several methods to get the best split they are

- Histogram-based methods
- Ignoring sparse inputs
- Gradient-based-one-side sampling
- Exclusive feature bundling

2.3 Overfitting

A model that has learned the noise instead of the signal is considered to over fit because it fits the training dataset but has a poor fit with new datasets. The signal is defined as the true underlying pattern that you wish to learn from the data, whereas noise is defined as the irrelevant information or randomness in a dataset. We need to reduce the overfitting the model to improve its performance. We can detect the overfitting when our model does much better on the training set than on the test set, and then there is a high chance of overfitting (Cawley & Talbot, 2010).

2.4 Overfitting Control Measures.

2.4.1 Cross-validation

In K-Fold validation, the data which the model needs to be trained is split into k-folds and the model is trained with all k-1 folds and it tested with the remaining fold. Similarly, it is done k number of times. In sections 5.3 the mean cross-validation scores and various metrics like F-1 score are compared to evaluate the classifier models for individual testbeds. To overcome the problem of overfitting I have used kfold = 5 and shuffle= true as parameters.

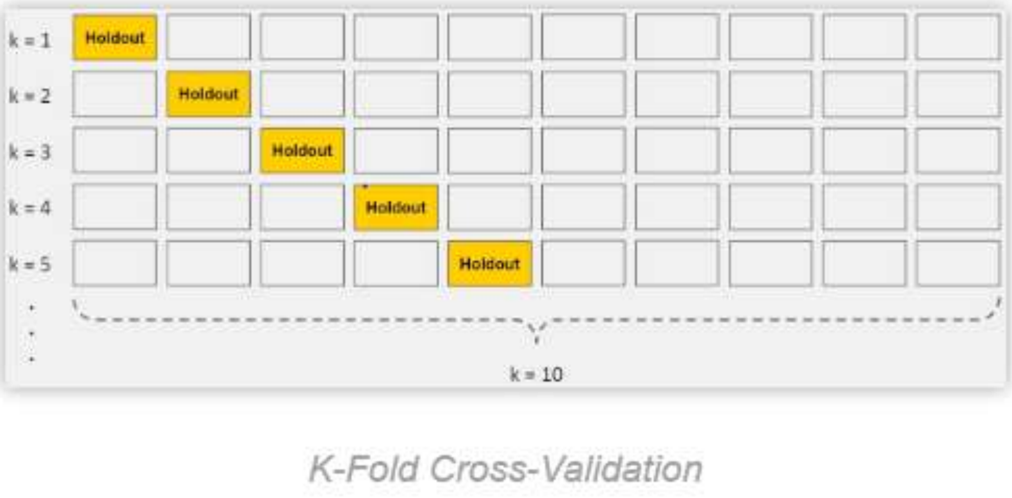

Fig 2.2: Working process of k-fold cross

Cross-validation allows us to tune hyperparameters with only your original training set. This allows you to keep your test set as a truly unseen dataset for selecting your final model.

2.4.2 Train with more data

This technique might not work always but if we train our model with more data then there is the chance that model will learn signal, but the data need to be clean and relevant. But if the data is not properly processed then the algorithm may degrade its performance.

2.4.3 Remove features

Removing the unnecessary features from the dataset will help to decrease the complexity of the model. To remove the highly correlated features I have used principal component analysis. I have discussed regarding principal component analysis in the Chapter 3.2.2 Feature Selection.

2.4.4 Early stopping

We can measure the performance of the model in each iteration when training the model iteratively. The performance of the model decreases after a certain iteration when the algorithm starts to overfit the model. So, we can stop training the model to avoid overfitting.

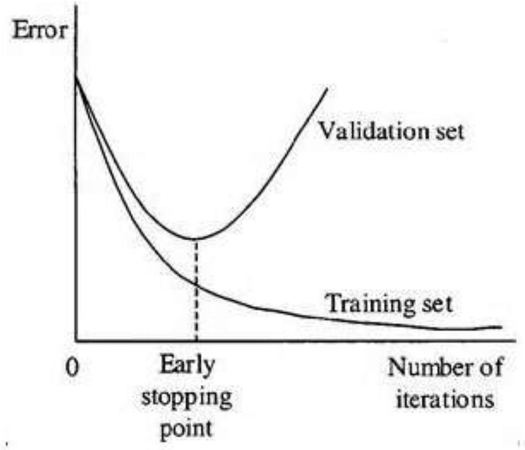

Fig 2.3 Detection of early stopping point

I have used early stopping in the LightGBM algorithm to avoid overfitting. I have used a parameter early_stopping_rounds=300. So, my model will be trained for at most 300 number of times and tries to find the early stopping point to avoid overfitting.

Chapter 3 -Implementation

This chapter includes an overview of the data, data preprocessing methods and implementation steps followed for the project.

3.1 Overview of the Data

Data place a very important role in any machine learning task. There are many factors which influence the success of the machine learning tasks such as quality of data, amount of data, etc.

To predict the correct poverty label of a household, I have used the dataset from Kaggle which Costa Rica Household Poverty Prediction. Each row in the household dataset contains 142 columns which describe the details of the household. To get a clear understanding of the dataset to refer to Appendix A. The below graph shows the distribution of data in the dataset.

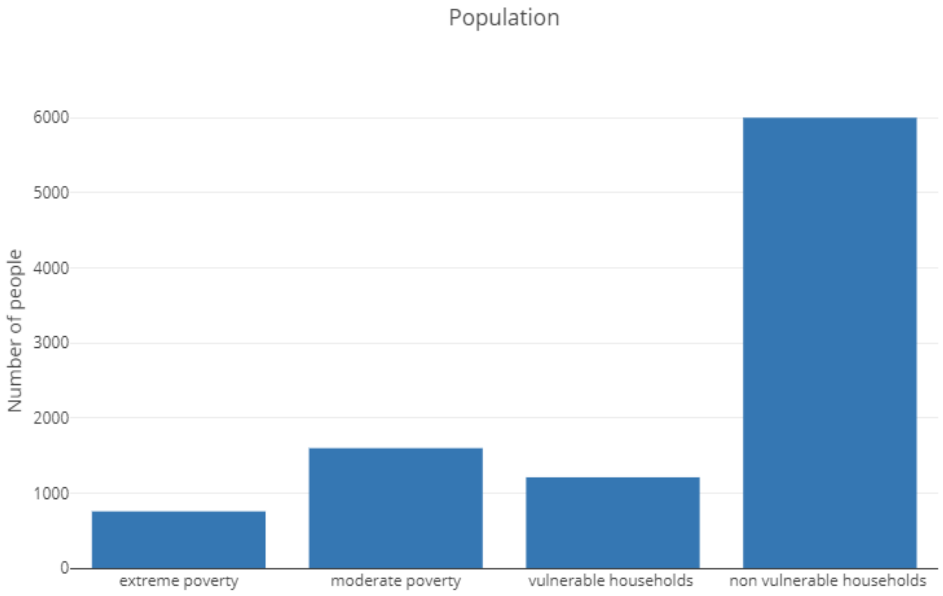

Fig 3.1 illustrates the distribution of labels in the dataset

3.2 Data Preparation

Data preparation plays a crucial role to yield better results in any machine learning task. It helps the dataset to be more suitable for machine learning task. This section describes the feature and other preprocessing techniques used to prepare the raw data to pass to the classifier to improve the results.

3.2.1 Data Preprocessing

As the data is collected from the real world it is a lot of missing values, noise and inconsistent entries. These factors will have a high influence in the results. To avoid incorrect outcomes, the raw data need to be processed. So, before passing the dataset to the machine learning classifier it needs to be cleaned and processed thoroughly. As it is a real data, there are some missing values for some columns such as v2a1 which represents the rent of the household it has a lot of missing values, so to fill the missing values the person who owns the house doesn’t pay the rent. In those cases, it is filled with the value 0 and in the other cases; I have filled it with the mean of the rents of the corresponding targets. But whereas in the case of test dataset we don’t have any target, so I have replaced it with the average of means of all the corresponding targets. In some other cases, the value of 0 is used as no and the value of 1 is used as yes. So, for better results, we need to replace it. Similarly, depending on the situation the values are replaced with suitable values.

I also removed some of the highly correlated columns ‘tamhog’, ’hhsize’, ’coopele’, ’female’, ‘hogar_total’, ‘area2’, removing some columns will decrease the complexity of the model. I have used python as the coding language for the project. Several libraries were used to implement the project. Few of which are pandas, scikit – learn, LightGbm, etc. Pandas is an open source library which is used mainly to get the data which is in the form of .csv and converting it into a dataframe and used as well for analysis and data manipulation task. Scikit learn is a free software machine learning library which is used for python language. It also consists of various clustering, regression and classification algorithms such as Random Forest, Gradient Boosting, K-Means and many more. In figure 3.1 most of the training data is in the poverty label 4. So, we need to avoid the overfitting of the data I have used cross-validation.

3.2.2 Feature Selection

Feature Selection plays an important role in improving the accuracy of the model. It is the process of selecting only a few features from the set of all features available. This process can improve the accuracy and precision of the model. There are many ways where we can implement this feature selection process. Some of the algorithms such as Random forest and LightGBM have an inbuilt feature importance selection. For other algorithms, we can use feature importance_ to get the importance of each feature. By implementing this process, we can improve the accuracy of the model and in some cases, it will be beneficial if the features are left as it is. I have used Principal Component analysis to remove the highly correlated features.

3.2.3 Principal Component Analysis (PCA)

Principal Component Analysis is the method of transferring highly correlated variables into a small number of variables which are known as principal components, which has a high variance in data. It uses a method known as Single value Decomposition in order to reduce the dimensionality of the features.

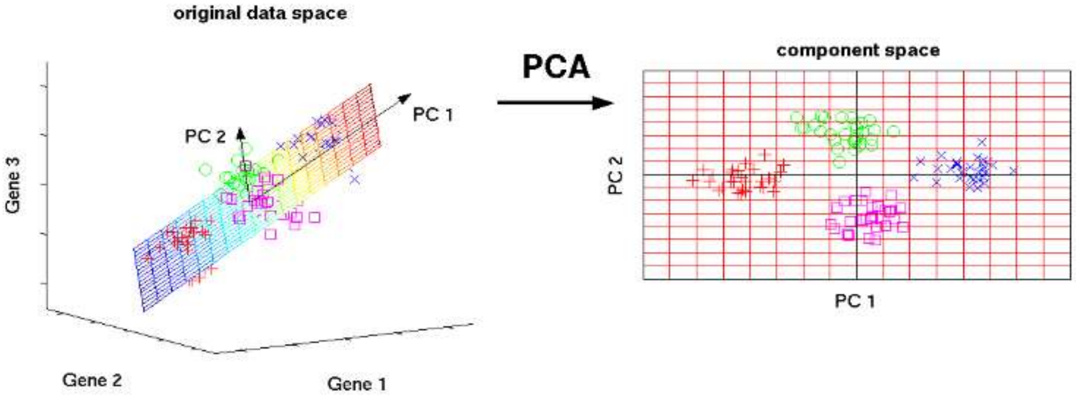

Fig 3.2 Working process of PCA to remove highly correlated variables

PCA is imported from sklearn.decomposition. To transform our data, we need to use PCA.fit_transform function and training data is passed as a parameter.

3.3 Implementation Steps

Below, the process flow along with implementation steps is shown. Figure 3.1 shows the process, starting from collecting the data from the Household poverty Dataset to predicting the poverty level of the household

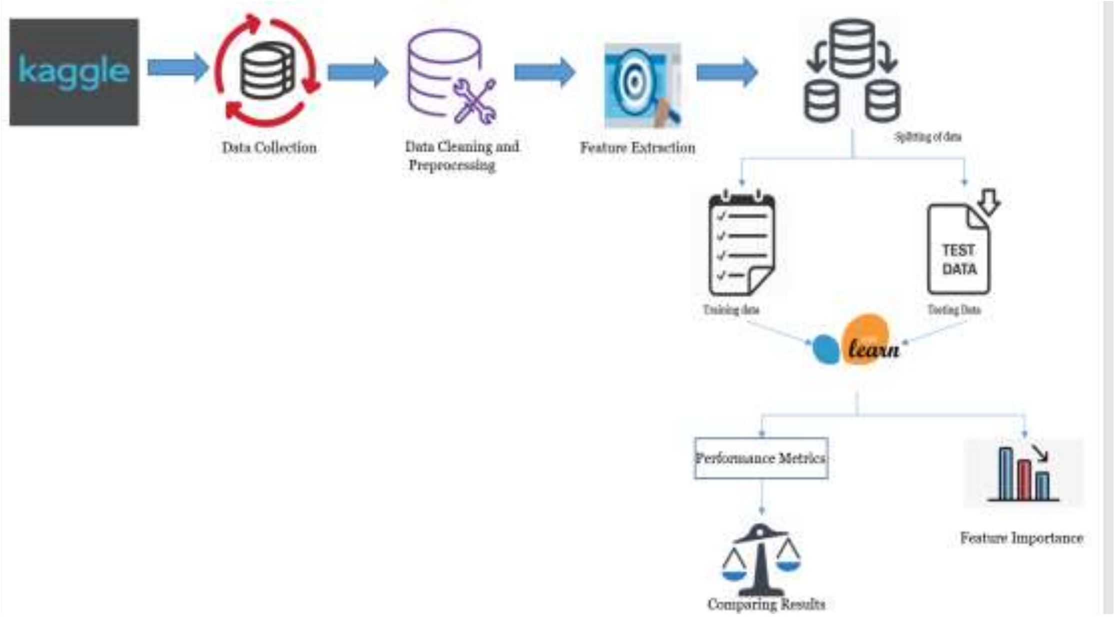

Figure 3.3 Workflow diagram of the project showing implementation steps

This is the step by step procedure for the working procedure of the project

- **Data Collection** - In this step I have collected the data from the kaggle in the csv and convert it into a suitable format for the classifier.
- **Data Cleaning and Preprocessing** - I have mentioned the steps followed in this stage in chapter 3.2.1
- **Feature Extraction-** Reducing the features will help to decrease the complexity of the models. Steps followed in this stage are mentioned in 3.2.2
- **Splitting the data** - I split the labelled data into seventy percent of it to training dataset and thirty percent of it to testing dataset.
- **Performance Metrics -** Used F1 score and recall as the performance metrics. Definitions of these metrics are mentioned in chapter 5.1

Chapter 4 - Experiment Design

This section illustrates the experiments conducted on the dataset. Different experiments that are conducted on the dataset using the various approaches are mentioned in sections 2.2

4.1 Training and Testing datasets

An important part of evaluating classification models is separating data into training and testing datasets. The training set is used to build the model and the test set is used to measure its performance. The data is divided into 70 % of the training data and 30 % of the testing data.

Total Records	33458
Training Records	9558
Testing Records	23900

Table 4-1 Household Data Distribution

4.2 Experiment Design

This section illustrates briefly about the experiments conducted on the dataset using various machine learning algorithms mentioned in Chapter 2. Scikit learn (Lars, et al., 2013) is a free software machine learning library which is used for python language. It also consists of various clustering, regression and classification algorithms these algorithms are used to predict the exact label of the household given. Here the training data is used to train the model. After the training of the model, it is evaluated with the test data which the model has not been trained.

4.2.1 Logistic Regression CV

Logistic Regression CV is implemented using the scikit-learn (Lars, et al., 2013) Library by importing the LogisticRegressionCV class from the sklearn.linear_model. Then the model to classify was fit on the training data using fit.predict_proba function which is used to predict the probabilities of testing data and the predict function is used to make predictions for household variables in test data. In order to implement the Logistic Regression and Cross-validation I have passed Training Set, Training Labels and LogisticCv as parameters to cross_val_score function which is imported from sklearn.model_selection (Lars, et al., 2013).

4.2.2 Random Forest

Random Forest was implemented using the library sklearn.ensemble (Lars, et al., 2013) and by importing RandomForestClassifier class. We need to pass all the features such as n_estimators, min_samples_leaf, max_features, n_jobs, etc before we fit the model and predicting the f1 score of the model.

4.2.3 LightGBM

LightGBM was implemented by importing lightgbm class. We also need to pass all the parameters such as learning_rate, n_jobs, n_estimators and objective before we fit the model and predicting the f1 score of the model.

Chapter 5-Results

This section has the results for the experiments described in Section 4.2.1 – 4.2.3 and the values of the metrics used to evaluate the performance of these models.

5.1 Evaluation Metrics

I used Macro F1 score and Accuracy as evaluation metrics to determine the performance of the algorithm (Murphy., 2002).

5.1.1 Macro F1 Score.

F1 score is the harmonic mean of precision and recall. Usually, the F1 score is optimal when we try to balance between precision and recall

$$\boldsymbol{F1 = 2 * \frac{precision \ * \ recall}{precision + recall}}$$

Eq 5.1.F1 Score Formula

5.1.2 Accuracy

Accuracy can be defined as the number of correct predictions made to the total number of predictions.

$$\boldsymbol{Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}}$$

Equation 5.2 Accuracy Formula

5.1.3 Macro F1 Score

Macro F1 score is slightly different when compared to F1 score. Here the precision and recall values are the average of precision and recall of the individual class.

P1=precision of class 1, P2=precision of class 2, P3=precision of class 3, P4= precision of class 4

$$\boldsymbol{Macro\ Precision = \frac{p_1\ + p_2 + p_3 + p_4}{4}}$$

Equation 5.3. Macro Precision

R1=recall of class 1, R2= recall of class 2, R3= recall of class 3, R4= recall of class 4

$$\boldsymbol{Macro\ Recall = \frac{R_1\ + R_2 + R_3 + R_4}{4}}$$

Equation 5.4. Macro Recall

5.1.3 Cross-Validation

To evaluate the model, I have used k-fold validation to evaluate the performance of the model. Cross-Validation can be implemented by importing cross_val_score function from sklearn.model_selection library (Lars, et al., 2013).

5.2 Experimental Results

The results of the algorithms described in chapter 4 are explained using the f1 score performance metric.

5.2.1 Logistic Regression

Evaluation Metric	Value
F1 Score	0.6825
Accuracy	0.7294

Table 5.1 Performance Metrics of Logistic Regression

The low F1 score for the logistic regression explains that as the data is nonlinear. Logistic regression cannot get the exact linear equation for the data. This resulted in the dip of F1 score value.

5.2.2 Random Forests

Here for the random forest, I have tried applying various techniques, such as down sampling the data, Up Sampling the data and limiting the features. I will include all the values I have got but the result was that I got the best F1 score with the default dataset.

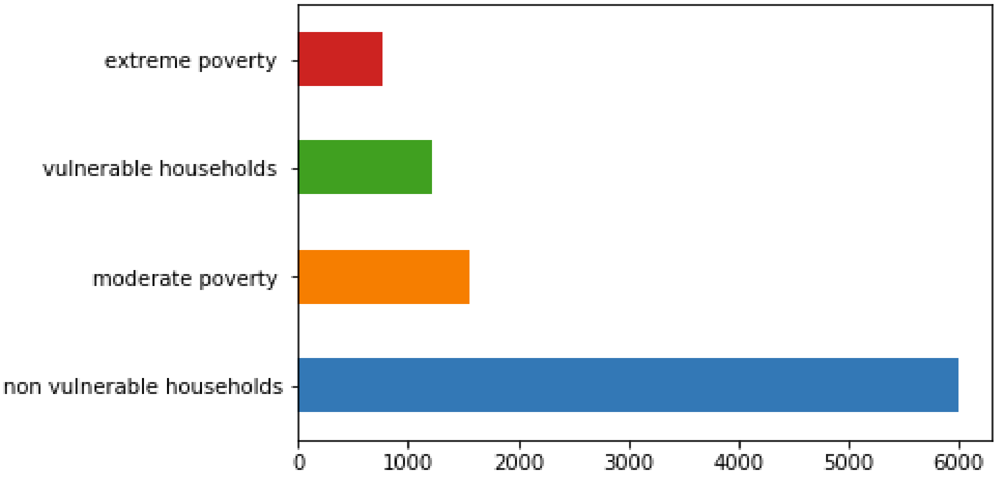

Fig: 5.1 Default dataset poverty level distributions

Evaluation Metric	Value
F1 Score	0.7881
Accuracy	0.8221

Table 5-2 Evaluation Metrics on Default Data- Random Forest

Class	F1 Score Value
Extreme Poverty	0.7848
Moderate Poverty	0.7448
Vulnerable households	0.7539
Non- vulnerable households	0.8687

Table 5-3 F1 values of each class on Default Data- Random Forest

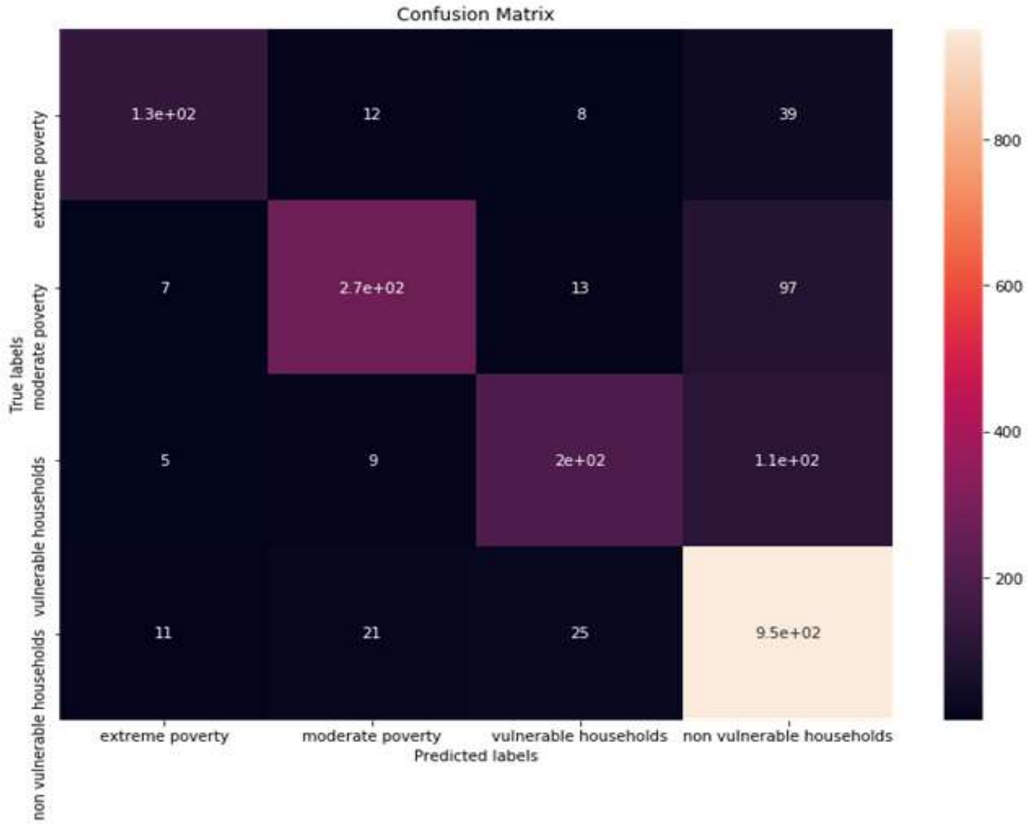

Fig 5.2: Confusion Matrix of Random Forest on default data

5.2.2.2 Oversampling the dataset

As we can see in the above case most of the values are tilted to label 4. When you train the model with the available data then there is a high chance that the model can be biased to the majority class. To avoid that scenario, I have duplicated the values in the minority class, to have equality among all the labels.

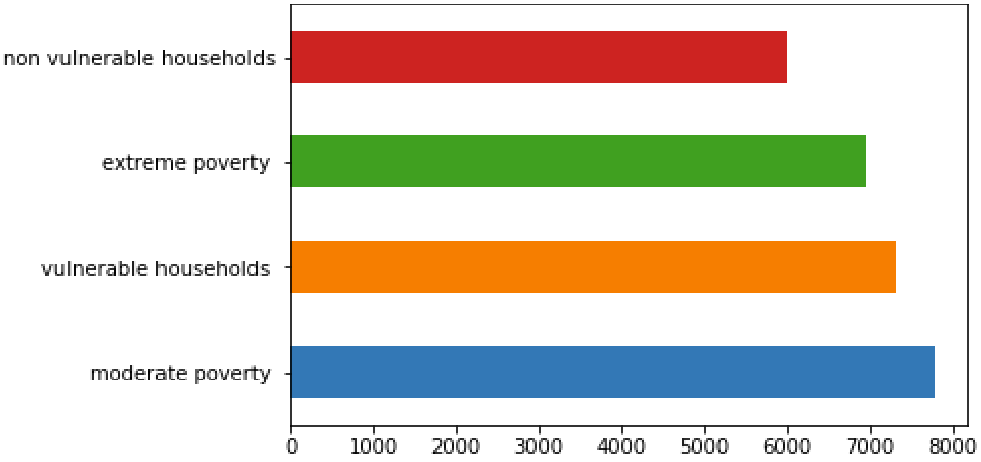

Fig: 5.3 Oversampling dataset poverty level distribution

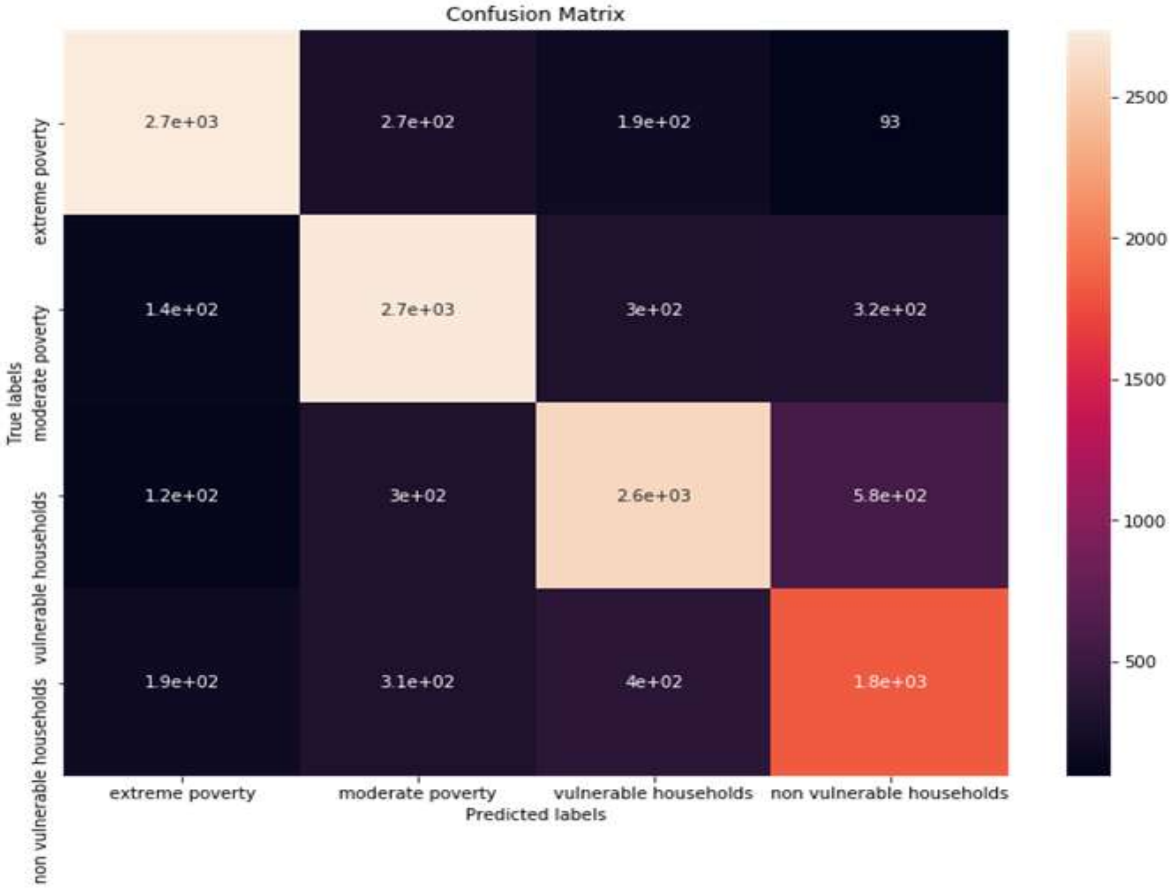

Fig 5.4: Confusion Matrix of Random Forest on Oversampled Data

The value of the f1 score of the random forest algorithm has been decreased. This is possible because the data has been increased; there can be a chance to increase the amount of noise in the data. Collecting more data will definitely increase the accuracy of the model.

Evaluation Metric	Value
F1 Score	0.7565
Accuracy	0.7611

Table 5-4 Performance Metrics on Oversampled Data- Random Forest

Class	F1 Score Value
Extreme Poverty	0.8533
Moderate Poverty	0.7698
Vulnerable households	0.7370
Non- vulnerable households	0.6657

Table 5-5 F1 values of each class on oversampled data- Random Forest

5.2.2.3 Downsampling the dataset

Downsampling the dataset means reducing the values of the majority class to have equality among the labels. For the dataset, I have reduced my majority class that is Label 4 to have the same number of rows for all the labels.

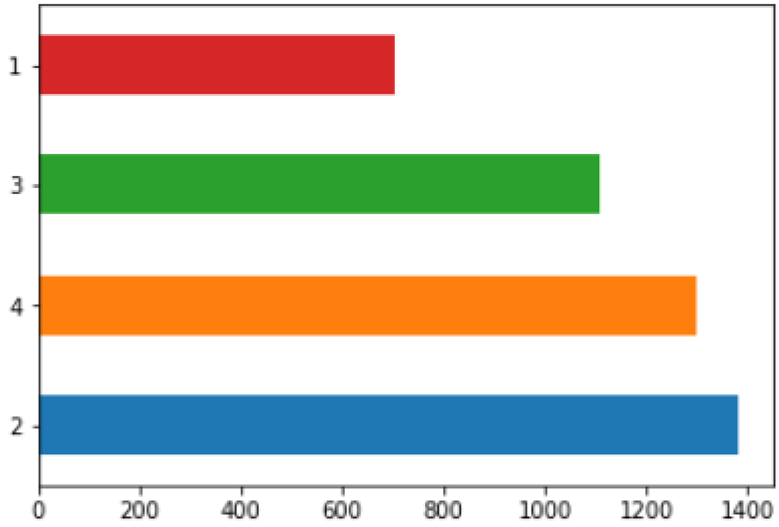

Fig: 5.3 Downsampling dataset poverty level distribution

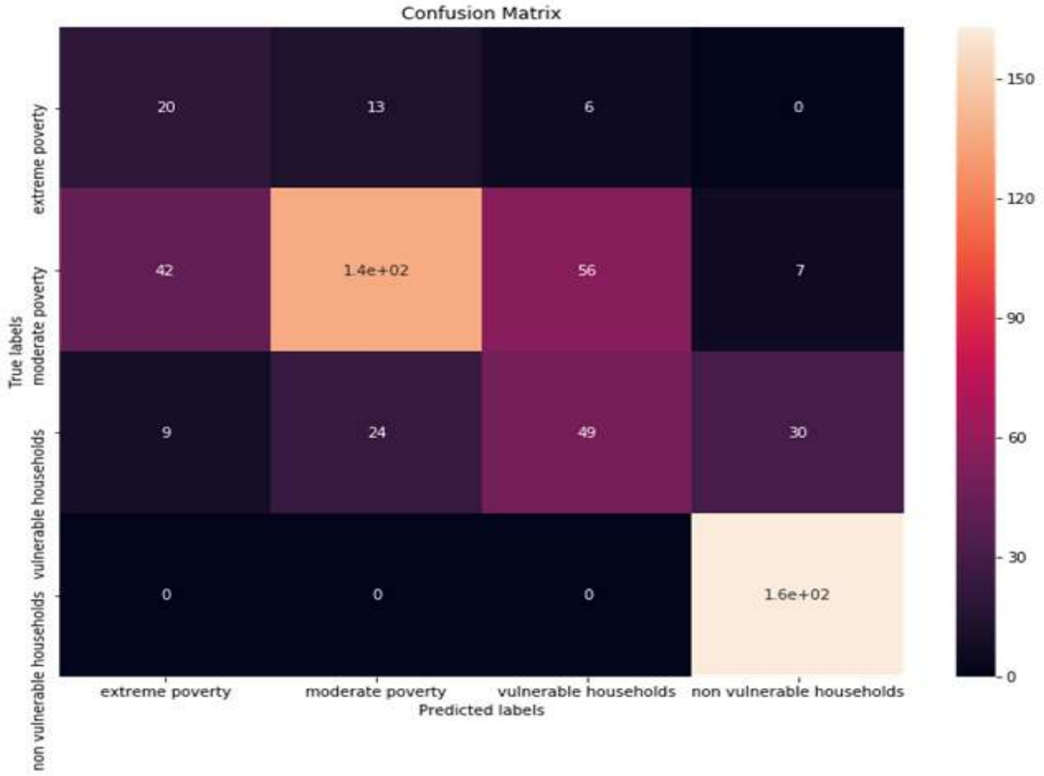

Fig: 5.4 Confusion Matrix of Random Forest on Downsampled Data

But in this case, as well the evaluation metric value did not improve. This can be possible because as we are reducing the majority class to about 25 percent of its original value. It is maybe losing all the important features this resulted in a reduction of F1 score.

Evaluation Metric	Value
Accuracy	0.6378
F1 Score	0.5719

Table 5-6 Evaluation Metrics on downsampled data- Random Forest

Class	**F1 Score Value**
Extreme Poverty	0.3965
Moderate Poverty	0.6313
Vulnerable households	0.4892
Non- vulnerable households	0.9041

Table 5-7 F1 values of each class on downsampled data- Random Forest

The parameters for the algorithm were: n_estimators=100, n_jobs=-1, min_samples_leaf=150. After performing a series of experiments, these values were opted to give the best F1 score value.

5.2.3 LightGBM

The parameters passed for this algorithm are base_learningrate=0.1, min_learning_rate=0.02, num_leaves=20, verbose="false". As the LightGBM performs the functions of both bagging and boosting the f1 score value improved a lot when compared to random forest

Evaluation Metric	**Value**
F1 Score	0.9843
Accuracy	0.9848
Macro F1 Score	0.862773

Table 5-8 Evaluation Metrics on LightGBM

Class	**F1 Score Value**
Extreme Poverty	1.0
Moderate Poverty	0.9745
Vulnerable households	0.9812
Non- vulnerable households	0.9864

Table 5-9 F1 values of each class on LightGBM

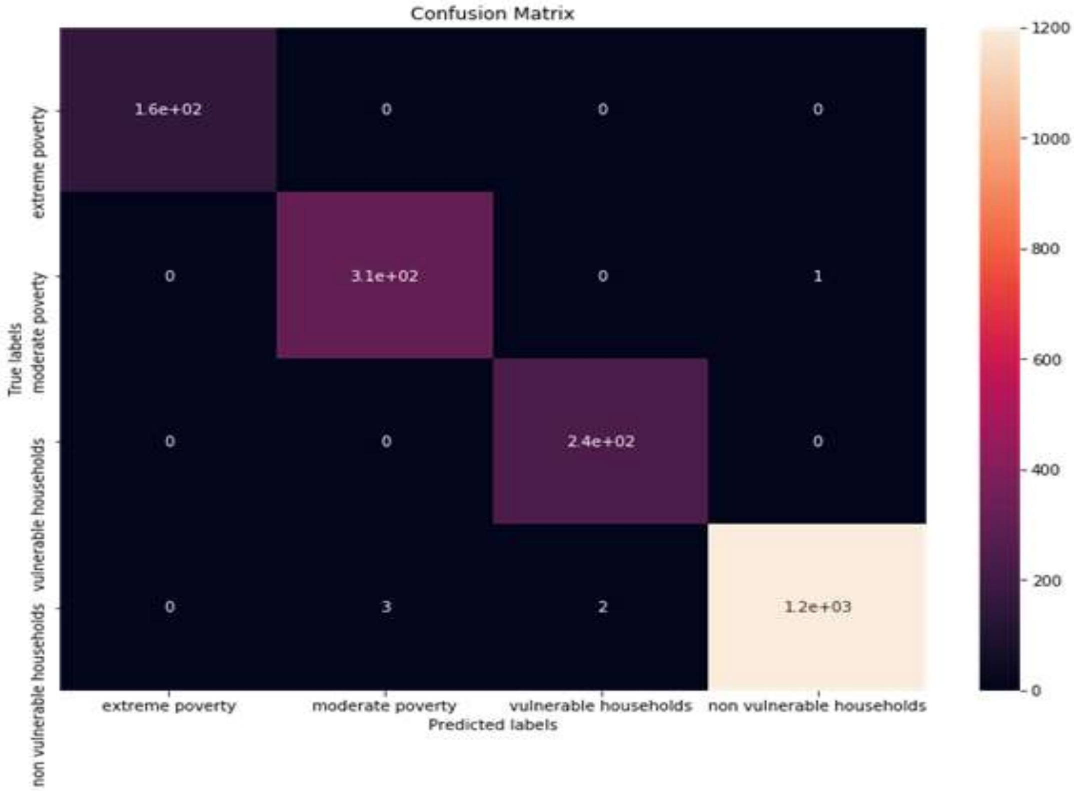

Fig: 5.5 Confusion Matrix of LightGBM

Chapter 6 - Summary and Future Work

This chapter draws conclusions and addresses the limitations of these approaches

6.1 Summary and Interpretation of Results

Based on the results of experiments shown in Chapter 5, it was clearly observed that LightGBM outperformed all other classification models. Logistic Regression has didn't perform well because as the data is not linear it was not able to generate the accurate equation. The next best alternative is Random Forest Classifier which had better results when compared to Logistic Regression.

6.2 Future Work of the Project

The data used for this comparatively less when compared to a data science project, having data of some bigger size will improve the results. We can also improve the model by hyperparameter tuning such as in the case of LightGBM we can tune the parameters by setting bagging_fraction, feature_fraction and having small max_bin we can improve the speed to the model. To improve the accuracy of the model we can use large max_bin and large num_leaves. In order to avoid the model from overfitting, we can train the model using Lasso Regression and Ridge Regression which are also known as L1 Regression and L2 Regression models (Ng, 2004). Regularization keeps model parameters close to some predetermined value or distribution (typically zero value) thereby reducing the difference between models trained on different realizations of the underlying distribution in the data. This difference is the variance error, and overfitting is the phenomena of a high variance error.

Bibliography

[1]Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems* (pp. 3146-3154).

[2]Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

[3] Pandey, S. M., Agarwal, T., & Krishnan, N. C. (2018, April). Multi-Task Deep Learning for Predicting Poverty from Satellite Images. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[4]Joshi, R. (2016, September 9). Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures. Retrieved from http://blog.exsilio.com/all/accuracy-precisionrecall-f1-score-interpretation-of-performance-measures/

[5]Lars, B., Gilles, L., Mathieu, B., Pedregosa, F., Mueller, A., Grisel, O., . . . Holt, B. (2013). Scikit Learn. Retrieved from http://scikit-learn.org/stable/

[6]N. J. Nilson. (2005). Introduction to Machine Learning. The MIT Press.

[7] T. M. Mitchell. (1997). Machine Learning. McGraw-Hill, Inc

[8] K. P. Murphy. (2002). Machine Learning: A Probabilistic Perspective. The MIT Press.

[9] F. Harrell. (2017, January 15). Classification vs. Prediction. Retrieved April 01, 2018, from Statistical Thinking: http://www.fharrell.com/post/classification/

[10] F. Pedregosa, et. al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 12, 2825-2830.

[11] Korkmaz, M., GÜNEY, S., & YİĞİTER, Ş. (2012). The importance of logistic regression implementations in the Turkish livestock sector and logistic regression implementations/fields. *Harran Tarım ve Gıda Bilimleri Dergisi*, *16*(2), 25-36.

[12] Cawley, G. C., & Talbot, N. L. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, *11*(Jul), 2079-2107.

[13]Ng, A. Y. (2004, July). Feature selection, L 1 vs. L 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning* (p. 78). ACM.

Appendix A

Household Dataset Description

Column Name	Description
v2a1	Monthly rent payment
hacdor, =1	Overcrowding by bedrooms
rooms	number of all rooms in the house
hacapo, =1	Overcrowding by rooms
v14a, =1	has a bathroom in the household
refrig, =1	if the household has a refrigerator
v18q	owns a tablet
v18q1	number of tablets household owns
r4h1	Males younger than 12 years of age
r4h2	Males 12 years of age and older
r4h3	Total males in the household
r4m1	Females younger than 12 years of age
r4m2	Females 12 years of age and older
r4m3	Total females in the household
r4t1	persons younger than 12 years of age
r4t2	persons 12 years of age and older
r4t3	Total persons in the household
tamhog	size of the household
tamviv	number of persons living in the household
escolari	years of schooling

rez_esc	Years behind in school
paredblolad =1	if predominant material on the outside
paredzocalo=1	if predominant material on the outside wall is socket
paredpreb=1	if predominant material on the outside wall is prefabricated or cement
pareddes=1	if predominant material on the outside wall is waste material
paredmad=1	if predominant material on the outside wall is wood
paredzinc=1	if predominant material on the outside wall is zinc
paredfibras=1	if predominant material on the outside wall is natural fibers
paredother=1	if predominant material on the outside wall is other
pisomoscer=1	if predominant material on the floor is mosaic, ceramic, terrazzo
pisocemento=1	if predominant material on the floor is cement
pisoother=1	if predominant material on the floor is other
pisonatur=1	if predominant material on the floor is natural material
pisonotiene=1	if no floor at the household
Pisomadera =1	if predominant material on the floor is wood
techozinc=1	if predominant material on the roof is metal foil or zink
techoentrepiso=1	if predominant material on the roof is fibre cement, mezzanine
techocane=1	if predominant material on the roof is natural fibres

techootro=1	if predominant material on the roof is other
cielorazo=1	if the house has ceiling
abastaguadentro=1	if water provision inside the dwelling
abastaguafuera=1	if water provision outside the dwelling
abastaguano=1	if no water provision
public=1	electricity from CNFL, ICE, ESPH/JASEC
planpri=1	electricity from the private plant
noelec=1	no electricity in the dwelling
coopele=1	electricity from cooperative
sanitario1=1	no toilet in the dwelling
sanitario2=1	toilet connected to sewer or cesspool
sanitario3=1	toilet connected to septic tank
sanitario5=1	toilet connected to black hole or latrine
sanitario6=1	toilet connected to another system
energcocinar1=1	no main source of energy used for cooking (no kitchen)
energcocinar2=1	the main source of energy used for cooking electricity
energcocinar3=1	the main source of energy used for cooking gas
energcocinar4=1	a main source of energy used for cooking wood charcoal
elimbasu1=1	if rubbish disposal mainly by tanker truck
elimbasu2=1	if rubbish disposal mainly by Botan hollow or buried
elimbasu3=1	if rubbish disposal mainly by burning
elimbasu4=1	if rubbish disposal mainly by throwing in an unoccupied space
elimbasu5=1	if rubbish disposal mainly by throwing in river, creek or sea
elimbasu6=1	if rubbish disposal mainly other
epared1=1	if walls are bad
epared2=1	if walls are regular
epared3=1	if walls are good
etecho1=1	if the roof is bad
etecho2=1	if the roof is regular
etecho3=1	if the roof is good
eviv1=1	if the floor is bad

eviv2=1	if the floor is regular
eviv3=1	if the floor is good
dis=1	if disabled person
male=1	if male
female=1	if female
estadocivil1=1	if less than 10 years old
estadocivil2=1	if the free or coupled union
estadocivil3=1	if married
Estadocivil4=1	if divorced
estadocivil5=1	if separated
estadocivil6=1	if widow/er
estadocivil7=1	if single
parentesco1=1	if household head
parentesco2=1	if the spouse/partner
parentesco3=1	if son/daughter
parentesco 4=1	if stepson/daughter
parentesco5=1	if son/daughter in law
parentesco6=1	if a grandson/daughter
parentesco7=1	if mother/father
parentesco 8.1	if father/mother in law
parentesco9=1	if brother/sister
parentesco10=1	if brother/sister in law
parentesco11=1	if other family members
parentesco12=1	if other non family members
Idhogar	Household level identifier
hogar_nin	Number of children 0 to 19 in household
hogar_adul	Number of adults in the household
hogar_mayor	of individuals 65+ in the household
hogar_total	of total individuals in the household
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)
meaneduc	average years of education for adults(18+)
instlevel1=1	no level of education
instlevel2=1	incomplete primary
instlevel3=1	complete primary
instlevel4=1	incomplete academic secondary level
instlevel5=1	complete academic secondary level

instlevel6=1	incomplete technical secondary level
instlevel7=1	complete technical secondary level
instlevel8=1	undergraduate and higher education
instlevel9=1	postgraduate higher education
bedrooms	number of bedrooms
Overcrowding	persons per room
tipovivi1=1	own and fully paid house
tipovivi2=1	own, paying in installments
tipovivi3=1	Rented
tipovivi4=1	Precarious
tipovivi5=1	other(assigned, borrowed)
computer=1	if the household has a notebook or desktop computer
television=1	if the household has TV
mobilephone=1	if the mobile phone
qmobilephone	No of mobile phones
lugar1=1	region Central
lugar2=1	region Chorotega
lugar3=1	region PacÃƒÂfico central
lugar4=1	region Brunca
lugar5=1	region Huetar AtlÃƒÂ¡ntica
lugar6=1	region Huetar Norte
area1=1	zona urban
area2=2	zona rural
age	Age in years
SQBescolari	escolari squared
SQBage	age squared
SQBhogar_total	hogar_total squared
SQBedjefe	edjefe squared
SQBhogar_nin	hogar_nin squared
SQBovercrowding	overcrowding squared
SQBdependency	dependency squared
SQBmeaned	the square of the mean years of education of adults (>=18) in the household
agesq	Age squared