Universität Zürich

Machine Learning for Economic Analysis

Replication Project: Peru Project

University of Zurich[UZH]

Andrianos Michail, Hatem Khrouf, Jessica Rey, Matthias Leuthard and Nina Reiser

Supervisors: Professor Damian Kozbur, Matteo Courthoud

March 9, 2021

Contents

List of Figures

List of Tables

1 Introduction

Anti-poverty programs play an important role in assisting poor families through governmental cash transfers in developing countries (Hanna and Olken, 2018). Targeting beneficiaries are usually based on household income. Unfortunately, household income in developing countries is often unobservable by the government. A large share of the population in developing countries is working in the informal sector and therefore outside the tax net (Hanna and Olken, 2018). Given this lack of income information, measures to proxy income more accurately are essential for targeting programs. This replication project is based on the work of Hanna and Olken (2018). They are using proxy-means testing (PMT) for household income predictions and the eligibility of cash transfers.

Proxy-means testing is a method often used by governments to obtain information of household income or consumption per capita through observable proxies such as home structure or the ownership of items. Estimation errors could lead to cash transfers to households above the poverty threshold (inclusion error) or poor household do not receive a transfer (exclusion error). Therefore, reducing proxy-means test errors is essential for government cash transfer programs with finite fund (Hanna and Olken, 2018).

Hanna and Olken (2018) use ordinary least squares (OLS) which is often used by governments to estimate income. Regressing the measure of poverty on the assets estimates the coefficient for the prediction of income. In this replication project we are using Machine Learning (ML) techniques to reduce inclusion and exclusion errors from imperfect targeting. ML is more flexible in its estimation compared to OLS which holds stronger assumptions such as linearity in parameters (James et al., 2013).

1.1 Aims and Objectives

The purpose of this project is to create a proxy-means test using the ML methods Random Forest and Neural Networks on the Peruvian Encuesta Nacional de Hogares (ENAHO) data set. We compute the Mean Squared Error (MSE) to assess the performance of the ML algorithms. In addition, we employ the Mean Absolute Error (MAE), an alternative evaluation metric which is more robust to outliers. We aim to reduce the MSE and MAE thus the above mentioned errors in order to produce higher predictive accuracy for the outcome variable, consumption per capita. The MSE's calculated in this project are compared to the MSE of Hanna and Olken (2018) to evaluate the prediction improvements of our methods. Further, the Figure *Social Welfare versus Inclusion Error* of Hanna and Olken (2018) paper is reproduced. We expect Random Forest and Neural Networks to improve the prediction of consumption per capita in the ENAHO data set. The key contribution of our work is the application of ML in an economically relevant way for the sake of predicting consumption per capita in Peru.

1.2 Overview of the Report

The outline of the project looks as follows. Section 2 provides a descriptive overview of the ENAHO data set used for this replication task. Thereafter, Section 3 discusses the methods used for our ML applications. This section is aimed to provide insight on our model and metric choices and presents the optimization procedure of the models. In the last part of Section 3 we focus on parameter tuning. Then the results of the Random Forest and Neural Network algorithm are provided in Section 4. Finally, we are concluding and discussing our findings in Section 5.

2 Explanatory Data Analysis

In this chapter we provide a descriptive overview of the data. For the input variables we apply a correlation heatmap and a plot which depicts the percentage of true values for each feature. The latter is applied in this fashion because all of the input variables are categorical which makes the data cleaning process, in particular the scaling and normalization part, superfluous. Furthermore, we show how the output variable "consumption per capita" is distributed on the training set and explain how we split the data.

2.1 Input Variables

Overall, the data set contains 73 binary input variables. Figure 1 displays the households percentage of true values for the predictors in an ordered way. In other words, the plot captures the percentage of households who are equipped with the corresponding features. The Figure shows that most of the input variables reach a percentage of true value of 80 percent and more which suggests that most of the proxies are held by at least 80 percent of the households. Furthermore, only few predictors are held by less than 50 percent of the households. The plot also shows that the maximum and minimum percentage values are attained by the features primary school and electricity revealing that almost all households have completed primary school where as less than 20 percent of the households have access to electricity.

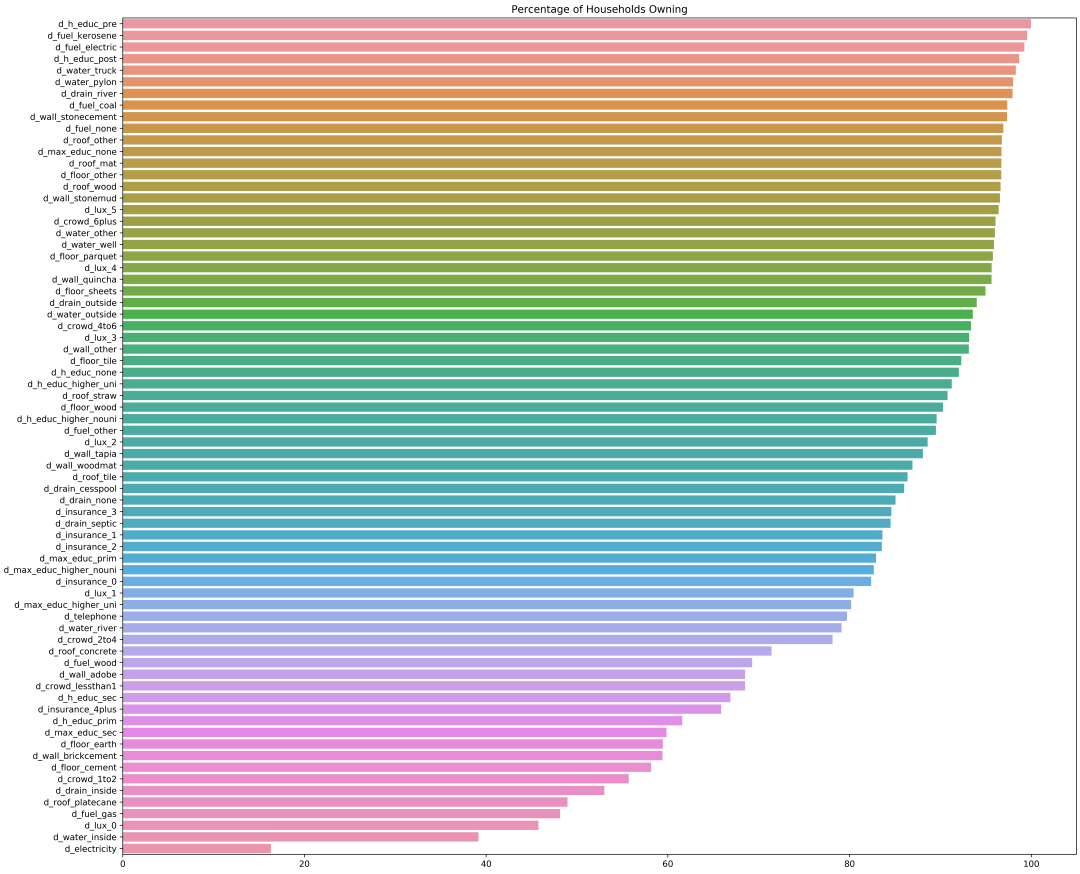

Figure 1: Household features percentages

In order to visualize co-movements between the features in our data set, we create a heatmap which is displayed in Figure 2. This map is based on the Pearson Correlation matrix and expresses correlations between the input variables (Kirch, 2008). A coefficient close to one suggests that two variables are highly correlated which is indicated in the heatmap as dark green, and can reach values up to -1 for highly uncorrelated variables (Kirch, 2008). Therefore, these regressors are more likely to be observed in conjunction. We only consider a subset of important features in the heatmap in order to ensure a good overview. It is worth noting that the heatmap in Figure 2 only shows correlations between variables and therefore we are unable to derive any causal statement from this matrix (Stock and Watson, 2015).

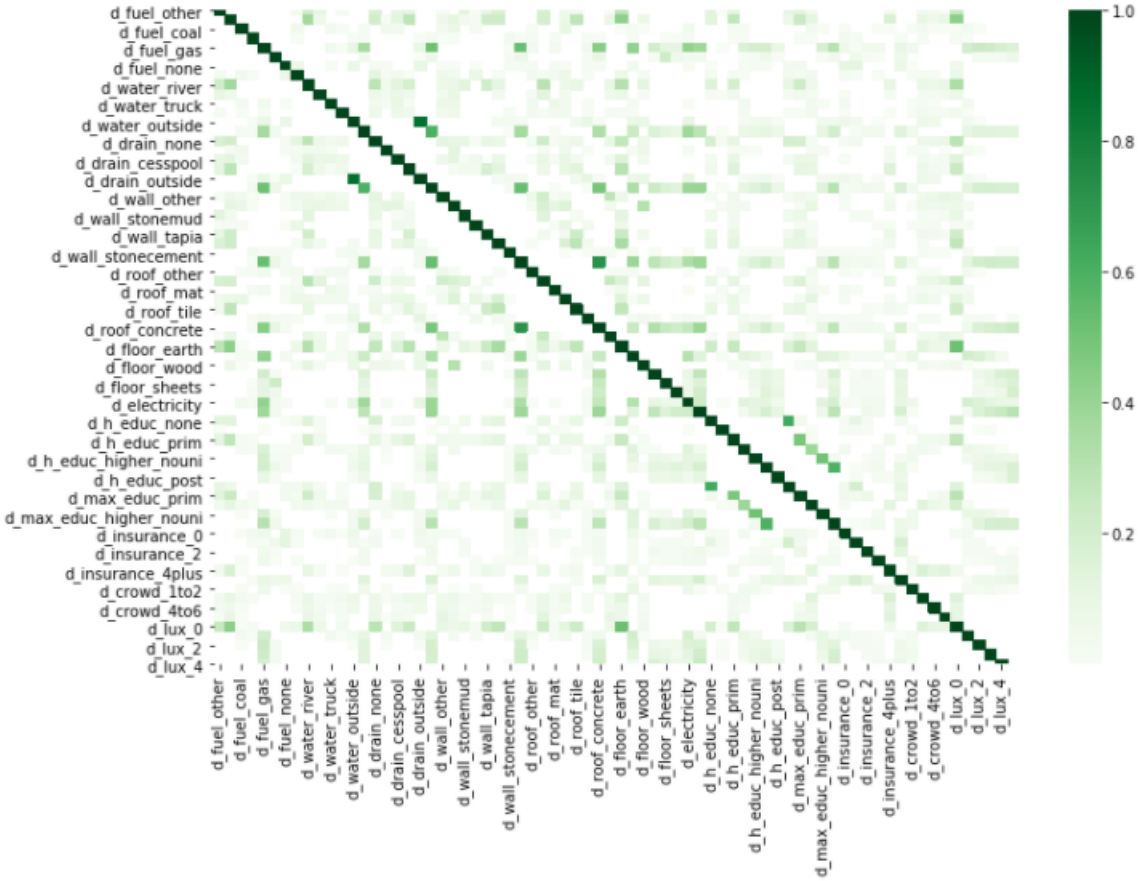

Figure 2: Pearson Correlation Heatmap (Positive Correlations only)

Figure 2 and further analysis suggests that the features fuel gas and drain inside are positively correlated with the majority of the features in the data. Due to its positive correlation across the board, it can be assumed that households that use gas and have a drain inside also own many other proxies in conjunction. Overall, the heatmap indicates that there is only a limited number of highly correlated input features in the data.

2.2 Output Variable: Consumption per Capita

As displayed in Figure 1, the distribution of consumption per capita on the training set is right skewed. Few people consume more than 1000 Nuevo Sols, the Peruvian currency, in a month. Most people consume around 500 Nuevo Sols or less per month. This means the surveyed households are highly concentrated to the bottom of the distribution and there isn't high variability in the distribution. Thus, identifying exactly which households should be included in a cash transfer program is hard to distinguish and further warrants the use of proxy means testing.

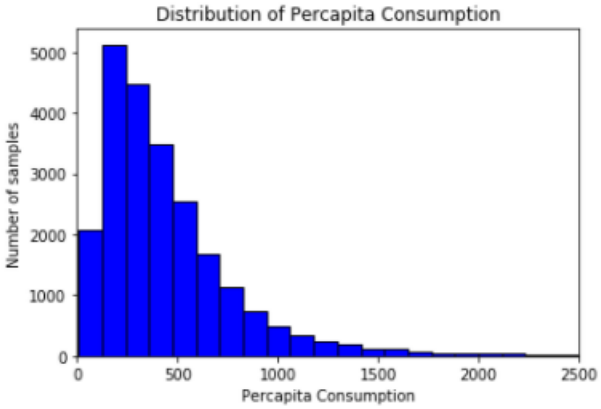

Figure 3: Percapita Consumption

2.3 Data Split

The data contains 23153 labelled observations and 23153 unlabelled data samples which is schematically shown in Table 1. Each data sample contains 73 binary features. To be able to validate and estimate the performance of the model, a portion of the training data was held to the side in order to use them as a labelled test data set. In order to train, validate and assess the performance of the supervised ML models, only the labelled observations are used.

Data Portion	**Labelled**	**Data Samples**
Training Data	Yes	20837
Hold Out Set	Yes	2316
Test Data	No	23153

Table 1: Segmented Data

As reported in Table 1, the defined Hold Out Set contains 2316 samples. This set is only used to estimate the performance of our final models and is not part of the training process in any other way.

3 Methodology

3.1 Evaluation Metrics

The models applied within this project, Random Forests and Neural Networks, can be computed to both regression and classification problems (Hastie et al., 2009; James et al., 2013). Since the scope of this project is to predict percapita consumption, a quantitative response taking on numerical values, we restrict our discussions to evaluation criteria used for regression tasks. More specifically, we compute the Mean Squared Error (MSE) and the Mean Absolute Error (MAE) to assess the performance of the ML algorithms applied in this project. Both measures indicate how well our predictions match the observed data (Bishop, 2006). More precisely, these evaluation metrics allow us to quantify how close the predicted response value for a given observation is to the true response value for that observation (James et al., 2013). Considering a regression model, this means that the MSE evaluates how well the regression line matches the observed data points (Bishop, 2006).

As displayed in Equation 1, the MSE is defined as the average squared distance between the predicted response $\hat{Y}_i$ and the true response Y_i or equivalently as the average squared error (Hastie et al., 2009; James et al., 2013). The MSE is rather small if the predicted responses are very close to the true responses, and therefore, it is aimed to minimise this expression (Hastie et al., 2009; James et al., 2013). However, it is not necessary how well the models work on the training data and therefore achieving a low training MSE is less of importance (James et al., 2013). Instead, we want to select the model that gives the lowest test MSE which is obtained when we apply our methods to previously unseen test data (James et al., 2013). Consequently, we compare the test MSE, estimated on the hold out sample from the ENAHO dataset, with the MSE presented in the paper of Hanna and Olken (2018) in order to evaluate our ML models.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{1}$$

The Mean Absolute Error (MAE) presented in Equation 2 is an alternative evaluation metric which we include for the sake of completeness. Similarly to the MSE, MAE quantifies the extent to which our predicted response is close to the observed value but it does so by computing the mean absolute distance between them instead of the mean squared distance (Bishop, 2006). As a result the MAE places much less emphasis on large errors and hence is more robust to outliers than the MSE (Bishop, 2006).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \hat{Y}_i| \tag{2}$$

3.2 Random Forest Regressor

In this chapter tree-based methods are introduced which provide a basis to construct more powerful prediction algorithms such as Random Forests. Moreover, the limitations of decision trees are discussed and demonstrate why an ensemble learning technique such as Random Rorest is more appropriate for our aim. Following Hastie et al. (2009) and James et al. (2013), regression trees[1] are typically constructed upside down. Starting at the top of the tree, the predictor space is sequentially segmented into distinct and non-overlapping regions such that less and also more homogeneous data points are aggregated under

[1] In the literature (Hastie et al., 2009; James et al., 2013) regression trees and decision trees are sometimes used interchangeably. In this project we use the term regression tree in order to emphasize that per capita consumption is a quantitative, non-categorical variable.

each branch.[2] Typically, a recursive binary splitting technique is used in order to construct the regression tree which is in the literature (Rokach and Maimon, 2005) sometimes referred to as a top-down greedy approach. In a nutshell, this technique aims to segment the regions and the respective cutpoints such that the resulting tree has the lowest residual sum of squares (RSS). Once the regions have been created, we make the same prediction for every observation that falls into that region. More specifically, the predicted response for an observation is typically given by the mean of the training observations in the region to which that test observation belongs.

Tree-based methods are simple, can be displayed graphically and are useful for interpretation even for non-experts (Hastie et al., 2009; James et al., 2013). However, recursive binary splitting as described above tends to produce too complex trees which suffer from high variance. This is due to their unlimited flexibility which leads to over-fitting. Consequently, trees can generate good results on training data but normally do not produce an optimal model when applied to unseen data in real world applications.[3] Several methods, such as bagging, Random Forests and Decision Tree pruning (e.g cost-complexity pruning or threshold pruning)[4] address the problem of over-fitted trees (Rokach and Maimon, 2005; James et al., 2013). In this project, we apply Random Forest, a simple, fast and non-linear learning algorithm which combines a large number of decision trees in order to produce a single consensus prediction. This method is more resistant to overfitting and therefore improves the predictive performance of trees substantially. As a result ensemble learning techniques such as Random Forests typically outperform regression trees in terms of prediction accuracy. Hence, we consider this method to be more suitable for the aim of this project since our focus is to achieve high predictive performance for per capita consumption and not on model interpretability.

Random Forests(RF) is a supervised ensemble learning method for classification and regression which combines multiple individual regression trees by means of bagging (Hastie et al., 2009; James et al., 2013). Bootstrap aggregation, or bagging produces several regression trees using different bootstrapped training sets and average the resulting predictions (Hastie et al., 2009; James et al., 2013). This procedure is schematically displayed in Figure 4 where 600 regression trees are built on bootstrapped training samples. Each tree in the Random Forest learns from a Random sample of the data points which are drawn with replacement (Breiman, 2001). Although bootstrap aggregation is a general-purpose procedure which can improve predictions for many different ML models, the method is particularly useful and frequently used for regression trees (Hastie et al., 2009; James et al., 2013). In this context the technique has been demonstrated to give impressive improvements in predictive accuracy (Hastie et al., 2009; James et al., 2013). This is because the increase in the number of trees[5] tends to decease the variance of the model without increasing the bias, therefore overcoming the over-fitting nature of regression trees (Breiman, 2001). In other words, averaging reveals the real structure that persists across data sets and noisy signals of individual regression trees cancel out (Breiman, 2001).

[2]The term branch refers to the segments of the trees that connect the nodes. The regions are denoted as terminal nodes or leaves and are, in line with the tree analogy, at the bottom of the tree while the points along the tree where the predictor space is split are denoted as internal nodes (Hastie et al., 2009; James et al., 2013).

[3]Another limitation of trees is the fact that they can be very non-robust which means that a small change in the data can lead to a large change in the final estimated tree (Hastie et al., 2009; James et al., 2013). However, we consider this problem to be less crucial in the context of our project since we face a given number of data points.

[4]More generally, this method considers depth of a tree as a hyperparameter which can be tuned through cross-validation. This approach reduces tree size and excludes those nodes with little predictive power. These techniques produce trees with fewer splits which have therefore lower variance and better interpretability at the cost of a little bias (Rokach and Maimon, 2005; James et al., 2013).

[5]Note that each individual tree has high variance, but low bias. (Hastie et al., 2009; James et al., 2013).

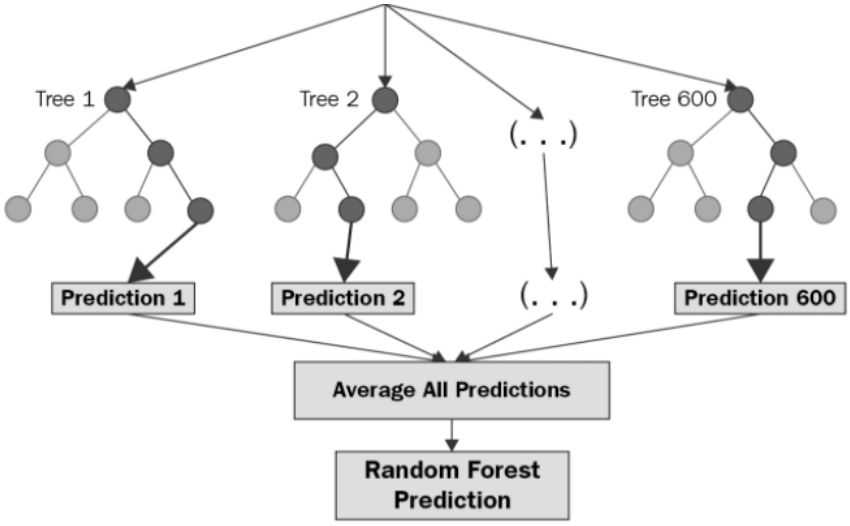

Figure 4: Random Forest Regressor

Source: https://levelup.gitconnected.com/random-forest-regression-209c0f354c84

In addition, each time a split in a tree is applied, Random Forest considers only a random subset of the predictors[6] and therefore reduces the correlation between different trees (Hastie et al., 2009; James et al., 2013). In Figure 4, this mechanism is indicated as dark grey path in each tree. The strength behind the Random Forest procedure is similar to that of a portfolio set-up where investments with low correlations form a portfolio is greater and more robust when diversified than that of the sum of its parts (Yiu, 2019). Similarly, uncorrelated trees can produce more accurate predictions because the trees protect each other from individual errors (Yiu, 2019). The property of Random Forest to use only a random subset of the predictors is crucial since otherwise the majority of the trees will select the strongest predictor in the top split (Breiman, 2001; James et al., 2013). This would produce similar bagged trees which are highly correlated to each other and averaging them will not lead to a substantial reduction in variance over a single tree (Breiman, 2001; James et al., 2013). In the context of our project this issue might be present since the heatmap presented in Section 2.1 indicates that the regressor electricity is highly correlated with other variables in our dataset. Random forests address this concern by forcing each split to consider only a subset of the predictors and therefore not only strong predictors such as fuel gas are selected but also other predictors will have more of a chance (Hastie et al., 2009; James et al., 2013).

3.3 Neural Networks - Multi Layer Perceptron Regressors

In this section the neural network or Multi Layer Perceptron (MLP) regressors are introduced. Artificial Neural Networks is a learning method with characteristics similar to biological Neural Networks in the human brain. It is a two-stage model which can be applied for both regression and classification (Hastie et al., 2009).

Neural networks usually consists of an input, a hidden and an output layer (see Figure 5). Each layer has a certain number of processing elements and signals are passed between these nodes. The input layer represents the original data and is through the hidden layer connected to the output layer. Neural

[6]Typically the number of predictors considered at each split is approximately equal to the square root of the total number of predictors (Hastie et al., 2009; James et al., 2013).

networks use linear weighted combinations of inputs and determine with a non-linear activation function the output variables. (Hastie et al., 2009; Ng and Soo, 2018; Zhang et al., 1998).

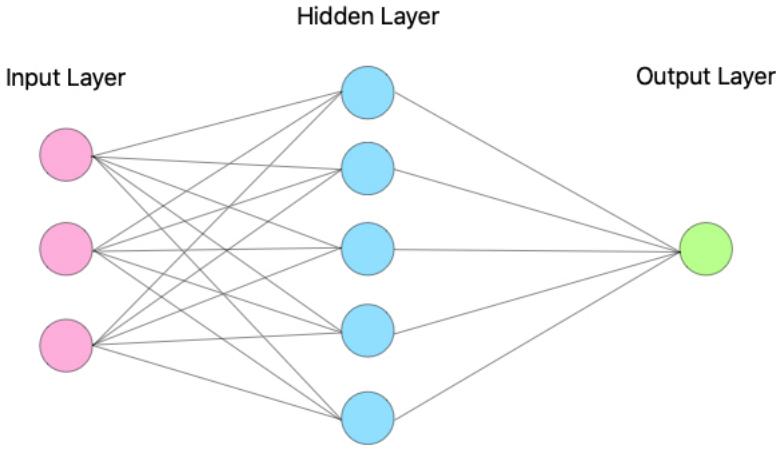

Figure 5: Neural Network Diagram

The input vector X with length p is weighted with w_m with $m = 1, .., M$. The inputs are weighted with unknown parameters which aim to make the model fit the training data well (Hastie et al., 2009). The features Z_m, derived in the middle layer, are unobservable and therefore also called hidden units. The hidden layer aims to capture the non-linearity in the data using the Relu activation function. Relu is a non linear function that outputs the input directly if it is positive, or 0 otherwise, as visualized in Figure 6 (Glorot et al., 2011). This function is calculated with the following formula:

$$Relu(x) = max(0, x) \tag{3}$$

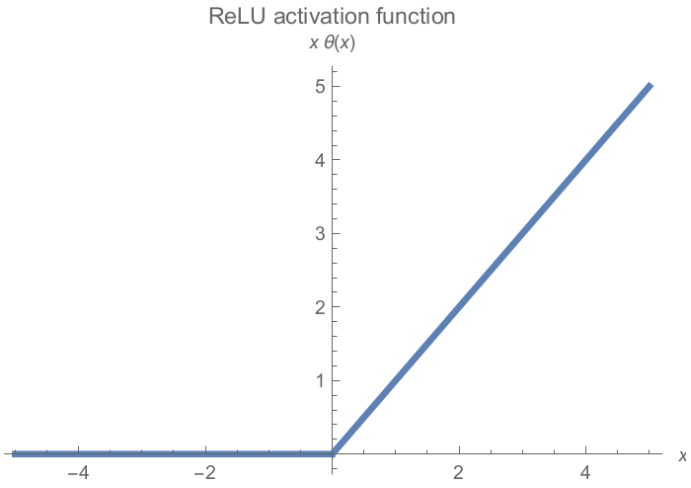

Figure 6: Visualisation of Relu

Source: https://miro.medium.com/max/970/1*Xu7B5y9gp0iL5ooBj7LtWw.png

The Relu function is used in this project since rectifying neurons are better than logistic Sigmoid neurons for training Neural Networks since rectifier units find better minima during training (Glorot et al., 2011).

The nodes of the output layer represent the dependent variable Y_t and are a linear combination of the derived features Z_m,

$$\begin{aligned} Z_M &= \sigma(\alpha_{0m} + \alpha_m^T X), m = 1, ..., M, \\ T_k &= \beta_{0k} + \beta_k^T Z, k = 1, ..., K, \\ f_k(X) &= g_k(T), k = 1, ..., K, \end{aligned} \tag{4}$$

where $Z = (Z_1, Z_2, ..., Z_M)$ and $T = (T_1, T_2, ..., T_M)$ (Hastie et al., 2009). For the final transformation of the output vector T as an output function $g_k(T)$, after Hastie et al. (2009) the identity function $g_k(T) = T_k$ is usually used in regressions.

Neural networks are trained by an iterative process. As a measure of fit the sum-of-squared errors,

$$R(\theta) = \sum_{k=1}^{K} \sum_{i=1}^{N} (y_{ik} - f_k x_i)^2 \tag{5}$$

is used for regression models. Back-propagation by gradient decent minimizes $R(\theta)$. In the process of back-propagation, each node receives and passes information only from and to units that share a connection. The Neural Networks model is learning to associate input variables to the correct output variables by iteration (Ng and Soo, 2018). Errors from the hidden and the output layer are used to compute the updated gradients (Hastie et al., 2009). In this project Adam, an algorithm for first-order gradient-based optimization of stochastic objective functions, is used (Kingma and Ba, 2014). An advantage of the Adam method is that it does not require a stationary objective, instead it works with sparse gradients and performs naturally step size annealing (Kingma and Ba, 2014).

3.4 Model & Predictions Procedure

The value desired to predict is the percapita consumption, however, in the training dataset, one additional value is available for regression, the lnpercapita consumption. Therefore, it was deemed appropriate, to train models for both of these values. Since calculating the exponent of a lnpercapita prediction results in percapita consumption, it was deemed appropriate to experiment with averaging multiple models and choosing the best predictions according to the holdout set.

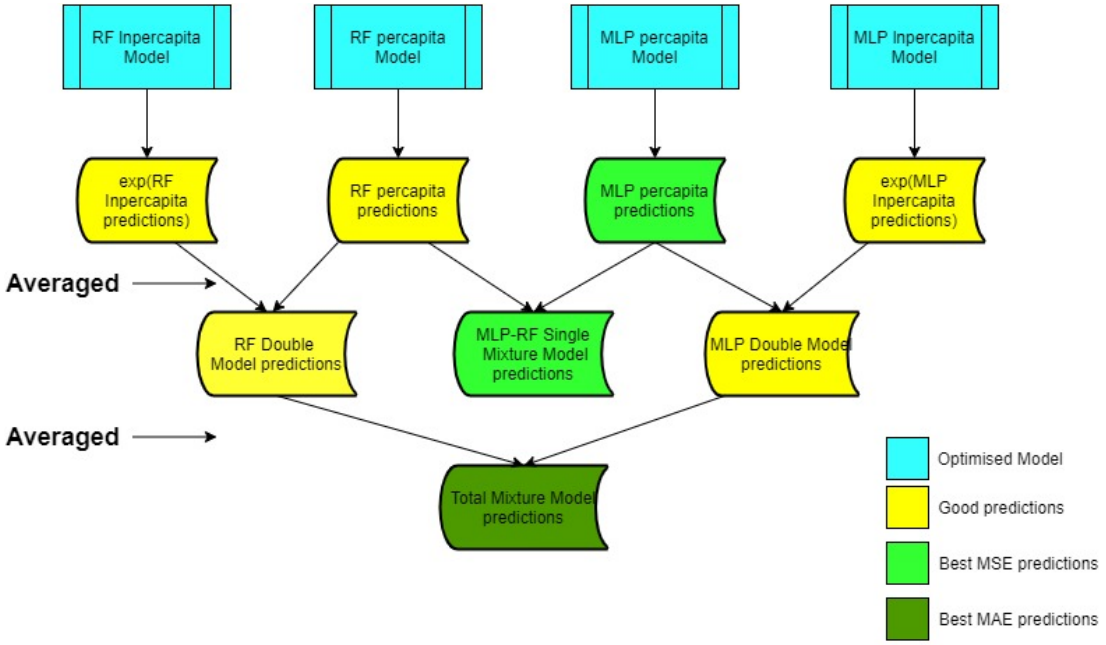

Figure 7: Predictions Model Diagram

As seen in Figure 7, multiple predictions have been created based on different models. The two best performing MSE models, the MLP per capita model and the MLP-RF Single Mixture model predictions performed best MSE as seen in Table 4.

Experimentation with averaging different predictions allowed to find a procedure that generates better regressors. As part of the assignment, 2 set of predictions are submitted, the ones with the better MSE, MLP-RF Single Mixture model and MLP percapita model respectively.

3.5 Optimisation

To optimise the ML models, there are hyperparameters which require to be tuned to provide optimal results. There are multiple methods available but due to the task in hand not being too computationally expensive, it was deemed appropriate to go with the classic Grid Search Cross Validation approach. The idea of grid search is that the model is trained with a portion of the data and each time tested through a validation set (which is chosen at each iteration) for all the possible combinations for a set of hyperparameters defined in the search space in a black box manner, where only minimising the one target value is important.

Through the Grid Search method, the value to optimise was set to be the MSE between the predictions and labels which is the metric the results are presented in at Hanna and Olken (2018). Each of the prediction models were both trained individually, but in the end, the best performing models end up having almost identical optimal hyperparameters.

3.5.1 Hyperparameter Search Space

```
#MLP Optimisation Space Search

param_grid_mlp = {
    'mlp_regr__activation': ['tanh', 'relu'],
    'mlp_regr__solver': ['sgd', 'adam'],
    'mlp_regr__early_stopping': [True, False],
    'mlp_regr__alpha': [0.0001, 0.001],
    'mlp_regr__learning_rate': ['constant','adaptive']}
```

```
#RF Optimisation Space Search

param_grid_rf = {'rf_regr__bootstrap': [True, False],
 'rf_regr__max_depth': [10, 20, None],
 'rf_regr__max_features': ['auto', 'sqrt'],
 'rf_regr__min_samples_leaf': [1, 2],
 'rf_regr__min_samples_split': [2, 5],
 'rf_regr__n_estimators': [100, 200]}
```

Figure 8: Hyperparameter Optimisation Search space

As seen in Figure 8, the search space was modest but sufficient enough to produce a well performing model for both the Neural Networks Regressor and Random Forest Regressors.

3.5.2 Best Performing Hyperparameters

Value	Bootstrap	Max Depth	Max Features	Min Leaf	Min Split	Estimators
percapita	FALSE	None	sqrt	2	2	200
lnpercapita	FALSE	40	sqrt	2	2	200

Table 2: Best Hyperparameters of RF Regressors from CV.

Value	Activation Function	Alpha	Early Stopping	Learning Rate	Solver
percapita	relu	0.001	FALSE	adaptive	adam
lnpercapita	tanh	0.001	TRUE	adaptive	adam

Table 3: Best Hyperparameters of MLP Regressors from CV.

As seen in Tables 2 and 3, the best hyperparameters for the models are very similar with minor differences for the two predictors, as the data trained is the same. One difference is the different activation function within the two different predictors. Further ablation study is not done within the scope of this assignment.

4 Results

4.1 Fitness Comparison

In this section we present our results from the per capita consumption prediction for each observation in the test sample. Table 4 displays the MSE and MAE for the Multi Layer Perceptron, Random Forest, the averaged models and the given OLS estimation for per capita consumption.

Prediction Model	Mean Squared Error	Mean Absolute Error
Multi Layer Perceptron Regressor Single Model	71298.921	155.688
Random Forest Regressor Single Model	72903.985	155.344
Multi Layer Perceptron Double Model	72270.128	152.261
Random Forest Regressor Double Model	75022.927	152.005
MLP-RF Single Mixture Model	70968.349	153.602
MLP-RF Total Mixture Model	72765.176	150.579
Given-OLS Predictions	82698.167	171.053

Table 4: Per capita consumption Performance

Comparing the first two regressor single model's, the MSE of the Multi Layer Perceptron is lower than the Random Forest's MSE, indicating a predicted response value closer to the true value. The significant lower MSE's for both ML techniques compared to the given OLS prediction validates our expectations regarding prediction improvements using ML techniques.

Averaging the per capita consumption predictions with the exponent of a logarithmic per capita predictions in a double RF and MLP model result in a slightly higher MSE but a lower MAE compared to the single model. The lowest MSE of 70968.349, thus best fit, is calculated with the MLP-RF single mixture model which averages the consumption per capita of the RF and MLP single model. The MLP-RF total mixture model performs the lowest MAE in comparison to all prediction models.

Training multiple models and experimentation with combinations of different prediction may lead to more accurate results. Our best model combines predictions from two different models (MLP-RF single mixture model), while the ones that combine exponent of lnpercapita predictions seem to not produce better results. Training separate models for lnpercapita and per capita appear to have better results. Table 5 presents the MSE and MAE calculation for the logarithmic per capita consumption. The low MSE of 0.191 of the MLP ln per capita regressor indicates a significant improvement of the prediction compared to the MSE of 0.928 of the OLS regressor.

Prediction Model	Mean Squared Error	Mean Absolute Error
MLP Inpercapita Regressor	0.191	0.340
Random Forest Inpercapita Regressor	0.192	0.340
Given OLS Predictions	0.203	0.456
MLP Inpercapita Exp-ed Regressor	75610.925	151.972
Random Forest Inpercapita Exp-ed Regressor	79100.031	151.670

Table 5: Ln per capita consumption performance

4.2 Social Welfare versus Inclusion Error

Lastly, we replicate the Figure *Social Welfare versus Inclusion Error* from Hanna and Olken (2018). The inclusion error, which translates to receiving a transfer even tough households are above the cutoff threshold, could still result in an increase in the total social welfare value. We are calculating the per-capita benefit amount of the included households for each level of the cutoff. The benefits households receive at each different cutoff are calculated in the training set and then applied to the 8.676 million of households in Peru in 2019 (Euromonitor International, 2020). A lower cutoff indicates higher benefits for the included households. We determine the social welfare with the same constant relative risk-aversion (CRRA) utility function as Hanna and Olken (2018):

$$U = \frac{\sum (y_i + b_i)^{(1-\rho)}}{1-\rho} \tag{6}$$

The variable y_i represents household i's income (per capita consumption) and b_i denotes the per capita benefits from the received transfer. The per capita consumption is our predicted value using the MLP model. It provides the highest prediction accuracy, indicated by the lowest MSE. The coefficient ρ captures the relative risk aversion of the households. A less negative value of the negative CRRA-utility function represents a higher utility. For the CRRA-utility calculation we are using $\rho = 3$, the same risk-aversion coefficient as Hanna and Olken (2018). According to Hanna and Olken (2018), the fixed transfer budget is \$274 million per year in Peru. We converted it to Nuevo Sols with the given information that \$30 are approximately 100 Nuevo Sols. The total social welfare is the sum of the individual CRRA-utility of the per capita income and the benefits received.

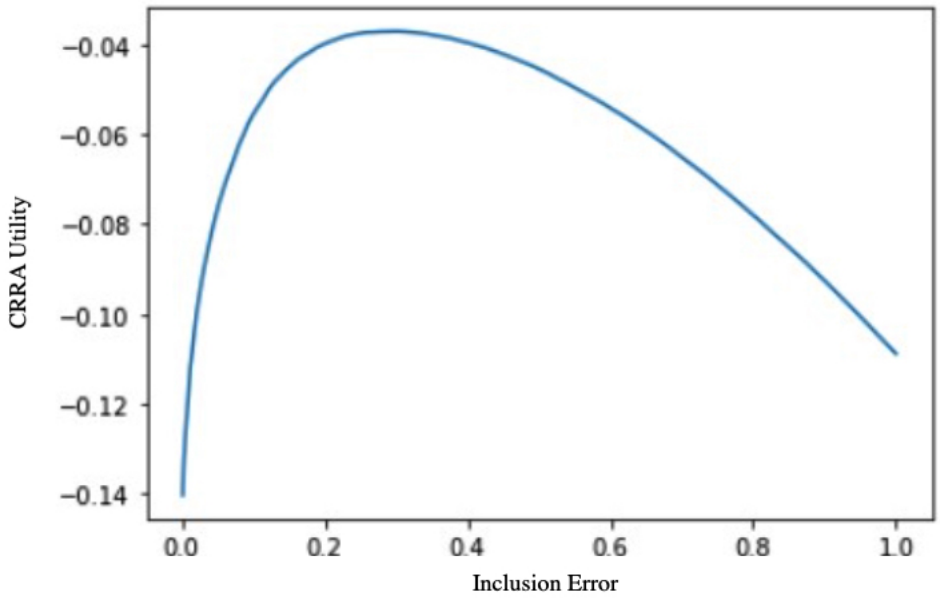

Figure 9: Welfare Analysis Inclusion Error generated with the predictions

In Figure 9 the social welfare for different levels of targeting are calculated with our own predictions of the consumption per capita from the MLP model. Setting the cutoff at the lower quantile of per capita consumption results in a lower inclusion error since fewer households which aren't actually poor are included. Setting the cutoff at the higher quantile of the per capita consumption on the other hand, induces a large inclusion error. The inclusion error is largest when households receive a universal basic income (UBI). The social welfare values declines with the increasing inclusions, where the inclusion error equal to one represents the universal basic income. The social welfare values are very low at the UBI level, indicating that even poor targeting is better than UBI.

5 Conclusion

Through this replication project, four ML models were trained and optimised, two Random Forests and two Multilayer Perceptron Regressors. As shown in the results Section 4, Neural Networks is the best performing algorithm. In fact, MSE was improved in all of the trained algorithms compared to the OLS, but the averaging of the Multilayer Perceptron and Random Forests prediction yielded the best results.

In light of our work on this replication project, and the results offered by Random Forests, Neural Networks and their mixtures, ML techniques offer promising improvements in proxy-means testing frameworks. This is of great importance for organisations with finite resources. Improving precision of PMT allows for household targeting to be more accurate, and thus funds spent more efficiently. In practice, the reduction in error and thus relative higher availability of funds could mean more households be included in a cash transfer program, funds be redirected or that cash transfers may be of higher value.

References

Bishop, C. M. (2006), *Pattern recognition and machine learning*, springer.

Breiman (2001), 'Random forests', *Machine Learning* **45**(1), 5–32.

Euromonitor International (2020), 'Peru Country Factlife'.
URL: *hhttps://www.euromonitor.com/peru/country-factfile*

Glorot, X., Bordes, A. and Bengio, Y. (2011), Deep sparse rectifier neural networks, Vol. 15, pp. 315–323.

Hanna, R. and Olken, B. A. (2018), 'Universal basic incomes versus targeted transfers: Anti-poverty programs in developing countries', *Journal of Economic Perspectives* **32**(4), 201–26.
URL: *https://www.aeaweb.org/articles?id=10.1257/jep.32.4.201*

Hastie, T., Tibshirani, R. and Friedman, J. (2009), *The elements of statistical learning: data mining, inference, and prediction*, Springer Science & Business Media.

James, G., Witten, D., Hastie, T. and Tibshirani, R. (2013), *An introduction to statistical learning*, Vol. 112, Springer.

Kingma, D. P. and Ba, J. (2014), 'Adam: A method for stochastic optimization', *arXiv preprint arXiv:1412.6980* .

Kirch, W., ed. (2008), *Pearson's Correlation Coefficient*, Springer Netherlands, Dordrecht, pp. 1090–1091.
URL: *https://doi.org/10.1007/978-1-4020-5614-7*$_2$569

Ng, A. and Soo, K. (2018), *Data science–was ist das eigentlich?!*, Springer.

Rokach, L. and Maimon, O. (2005), 'Top-down induction of decision trees classifiers-a survey', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **35**(4), 476–487.

Stock, J. H. and Watson, M. W. (2015), *Introduction to econometrics*.

Yiu, T. (2019), 'Understanding Random Forest'.
URL: *https://towardsdatascience.com/understanding-random-forest-58381e0602d2*

Zhang, G., Patuwo, B. E. and Hu, M. Y. (1998), 'Forecasting with artificial neural networks:: The state of the art', *International journal of forecasting* **14**(1), 35–62.