New Generation Computing (2023) 41:155–184
https://doi.org/10.1007/s00354-023-00203-8

NEW GENERATION COMPUTING

A Comparative Analysis of Multidimensional COVID-19 Poverty Determinants: An Observational Machine Learning Approach

Sandeep Kumar Satapathy[1] · **Shreyaa Saravanan**[1] · **Shruti Mishra**[1] · **Sachi Nandan Mohanty**[2]

Received: 1 April 2022 / Accepted: 27 December 2022 / Published online: 1 February 2023

Abstract

Poverty is a glaring issue in the twenty-first century, even after concerted efforts of organizations to eliminate the same. Predicting poverty using machine learning can offer practical models for facilitating the process of elimination of poverty. This paper uses Multidimensional Poverty Index Data from the Oxford Poverty and Human Development Initiative across the years 2019 and 2021 to make predictions of multidimensional poverty before and during the pandemic. Several poverty indicators under health, education and living standards are taken into consideration. The work implements several data analysis techniques like feature correlation and selection, and graphical visualizations to answer research questions about poverty. Various machine learning, such as Multiple Linear Regression, Decision Tree Regressor, Random Forest Regressor, XGBoost, AdaBoost, Gradient Boosting, Linear Support Vector Regressor (SVR), Ridge Regression, Lasso Regression, ElasticNet Regression, and K-Nearest Neighbor Regression algorithm, have been implemented to predict poverty across four datasets on a national and a subnational level. Regularization is used to increase the performance of the models, and cross-validation is used for estimation. Through a rigorous analysis and comparison of different models, this work identifies important poverty determinants and concludes that overall, Ridge Regression model performs the best with the highest R^2 score.

Keywords Poverty · Prediction · Multidimensional · Machine learning · Regression · Feature selection

✉ Shruti Mishra
shrutim2129@gmail.com

Extended author information available on the last page of the article

1 Introduction

The onset and continuation of the global pandemic COVID-19 saw a rise in extreme global poverty for the first time in 20 years, according to reports published by the World Bank [1]. Without an adequate response to this crisis, the brunt of the cumulative effects of the pandemic, climate change and armed conflict in the economic and social spheres have fallen on the underprivileged section of society who are unable to make ends meet. Studies focusing on the effect of poverty on middle-income countries have shown that these countries are more likely to be significantly affected [2]. Thus, it is important to find a method to focus on predicting poverty in developing countries. Predicting poverty comes with its set of complications. First, identifying poverty and regions in need is an extenuating task that requires patience and precision. Next, developing models that can efficiently aid in poverty prediction is not simple; a variety of factors must be taken into consideration, from the indicators to the levels of regional deprivation, which varies across different countries. Furthermore, the collection of household data is expensive and time-consuming. Poverty also has several definitions and angles, but this paper focuses on multidimensional poverty. Traditional methods of measuring poverty have been replaced with machine learning techniques, which can save time and transform the way that poverty estimation is approached.

Applying regression models to poverty data can provide precise predictions that would otherwise not be feasible. Machine learning algorithms have been used to predict poverty from household survey data and have been extended to using deep learning for mapping poverty [3, 4]. This paper introduces the application of machine learning methods to recent data and incorporates feature selection and validation techniques to poverty prediction. The work aimed to predict poverty on a national and subnational level to paint a lucid image of multidimensional poverty.

2 Related Work

Alkire et al. [5], made an analysis of the global multidimensional poverty and COVID-19 which is at risk since a decade of progress. They evaluated the potential impact of the COVID-19 and responses on global poverty using different poverty index. A harmonized trend for few countries was provided by Alkire et al. [6], where it was found that there was a significant reduction of multidimensional poverty due to the effect of COVID-19. This was measured by overlapping deprivations in the health and education domains. Anderson et al. [7] found that the policies measured that were implemented based on the epidemiological characteristics of the pandemic had a great context of uncertainty and their effect on the society were extremely large. Tavares et al. [8] proposed two indexes for measuring and indexing the vulnerability of the COVID-19 poverty based on the

evidence from a fuzzy multidimensional perspective. Huang et al. [9] examined the poverty trap through a multidimensional energy and its determinant in typically six provinces in China. They used the household-scale variables and community-scale variables for analyzing the effect.

3 Data Collection and Pre-processing

In the section, the data collected and used for this research will be discussed, along with the methods used to clean the data to prepare it for further steps.

A) Data Acquisition

The datasets were acquired from the Oxford Poverty and Human Development Initiative (OPHI)'s Multidimensional Poverty Index (MPI) reports [10]. The global MPI is an international measure of multidimensional poverty that covers over 100 developing countries and overcomes the limitations of household surveys [11]. The index assesses poverty intensity at an individual level using several socioeconomic indicators ranging from health to education and income. The results of the reports indicate disaggregation by age group, rural/urban areas, and subnational regions, with multiple poverty cut-offs. Four datasets were used for analysis and prediction: national data for 2019 and 2021, and subnational data for 2019 and 2021. These datasets were formed from the "Censored Headcount" MPI reports, which contain data on the intensity of deprivation for different indicators falling under the health, education, and living standards categories. The data are updated every year and consist of the latest developments across the globe.

Table 1 Important features of MPI dataset

Attribute	Description	Data type
Country	Country Name	Object
Multidimensional poverty index (MPI)	MPI value of the country (ranges from 0 to 1)	Float
MPI of the region	MPI value of a particular region in a country (ranges from 0 to 1)	Float
Nutrition	Percentage of population deprived in nutrition	Float
Child Mortality	Percentage of population deprived in child mortality	Float
Years of schooling	Percentage of population deprived in years of schooling	Float
School attendance	Percentage of population deprived in school attendance	Float
Cooking fuel	Percentage of population deprived in cooking fuel	Float
Sanitation	Percentage of population deprived in sanitation	Float
Drinking water	Percentage of population deprived in drinking water	Float
Electricity	Percentage of population deprived in electricity	Float
Housing	Percentage of population deprived in housing	Float
Assets	Percentage of population deprived in assets	Float

B) Dataset Description

The National datasets consist of 19 columns inclusive of 10 MPI indicators, and the Subnational datasets consist of 26 columns inclusive of 10 MPI indicators. Table 1 presents information about important attributes present in the datasets.

Other than the indicators mentioned in Table 1, the Regional datasets contain extra attributes describing the population for a particular region in 2016 and 2017 for the 2019 report, and the population for the years 2018 and 2019 for the 2021 report.

C) Data Cleaning

As part of the exploratory data analysis conducted, missing data was handled. The rows containing null values were transformed using the Python library pandas. The null values were imputed with the value 0.0 as there were a great number of rows consisting of null values in the subnational datasets, due to the scarcity of data from developing regions.

4 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is an important process during data analysis which allows the discovery of trends and patterns through maps and graphs. Since the MPI data consist of multiple variables, it is categorized under multivariate analysis, and a few techniques have been demonstrated below:

A) Feature Correlation

Correlation is a statistical measure used to represent the linear relationship between two variables. The correlation values help with feature selection as the features with high values of correlation can be excluded to avoid multi-collinearity. A correlation heatmap is a graphical representation of a correlation matrix that represents the correlations between different variables. Figure 1 shows the correlation heatmaps for the National datasets for the years 2019 and 2021, and Fig. 2 shows the graphs for the Subnational datasets for 2019 and 2021.

From Fig. 1, it is observed that certain independent variables like nutrition and school attendance are strongly correlated with variables like housing and electricity. MPI is strongly correlated with a majority of the independent variables due to its dependent nature. Since MPI is made up of a combination of indicators, it follows that it is dependent on variables like nutrition, drinking water, etc.

The correlation heatmaps for the subnational datasets differs significantly from the national datasets due to the introduction of the population attributes. From Fig. 2, the population attributes are negatively correlated with the MPI indicators for the 2019 dataset. The health, economic and living standards attributes are observed to be closely correlated for both 2019 and 2021.

Fig. 1 Feature correlation heat maps for National Datasets (left: 2019, right: 2021)

Fig. 2 Feature correlation heat maps for Subnational Datasets (left: 2019, right: 2021)

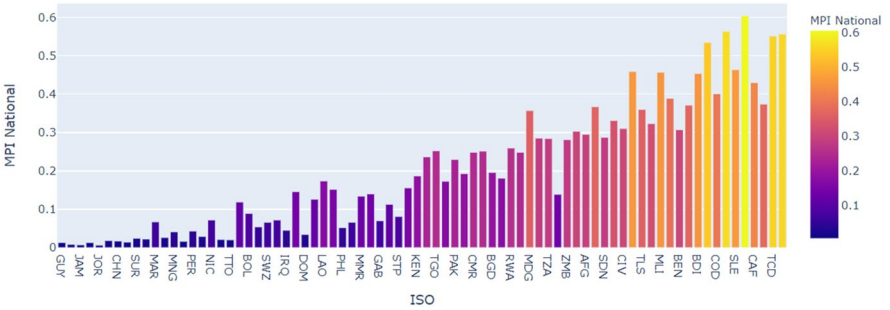


Fig. 3 Bar chart with MPI national values for all countries in National dataset

Fig. 4 Countries with the highest poverty headcount ratio

	Country	Total Headcount Ratio
81	South Sudan	176.500000
18	Chad	158.210000
79	Somalia	152.820000
67	Niger	150.090000
78	Sierra Leone	148.330000
17	Central African Republic	147.990000
50	Liberia	145.360000
30	Ethiopia	142.670000
13	Burkina Faso	140.460000
22	Congo, Democratic Republic of the	136.780000

B) Graphical Visualizations and Inferences

The national MPI values of the countries in the National dataset with 2021 data have been visualized in the form of a bar chart in Fig. 3. It can be seen that South Sudan has the highest MPI value, indicating a greater degree of deprivation.

The total headcount ratio of the MPI dataset indicates the number of people who are considered to be living in poverty based on several indicators. This is a measure of how much of the population contributes to national poverty and the ten countries with the highest headcount ratio are shown in Fig. 4. The country with the highest national poverty ratio is South Sudan, followed by Chad and Somalia.

5 Machine Learning Models

This section of the paper delves into the training and implementation of the machine learning algorithms. A flow diagram of the entire process is depicted in Fig. 5.

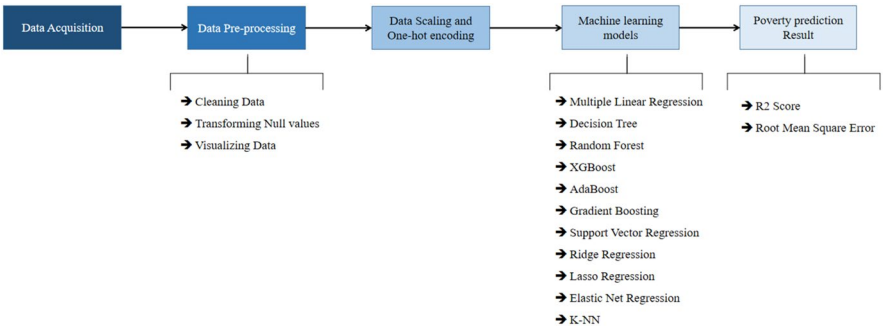


Fig. 5 Flowchart depicting a high-level view of the process to predict poverty using machine learning

A) Data Sampling

Splitting the data into training and testing datasets is an essential step in any machine learning research. This was done in a 70:30 ratio respectfully. The models were trained to predict the MPI of the country for the National datasets and MPI of the region for the Sub-national datasets.

B) Models

The performances of various state-of-the-art regression models were analyzed, namely Multiple Linear Regression, Decision Tree Regressor, Random Forest Regressor, XGBoost, AdaBoost, Gradient Boosting, Linear Support Vector Regressor (SVR), Ridge Regression, Lasso Regression, ElasticNet Regression, and K-Nearest Neighbor Regression.

a) Multiple Linear Regression

Owing to its simplicity, linear regression [12] is a simple and powerful algorithm for real-life data. Multiple linear regression [13] is an extension of the popular linear regression algorithm where multiple features (x_{1i}, x_{2i}, x_{3i},…,x_{pi}) are used to predict the target variable (y_i), which is MPI in this case. The reason why it is called 'linear' is that there is an assumption that that response variable is and the explanatory variables have a linear relationship, i.e., the target variable can be expressed as a linear combination of the features. Equation 1 represents multiple linear regression.

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \ldots + \beta_p x_{pi} + e \tag{1}$$

where, y_i is the dependent variable, and $\beta_{0..p}$ are the coefficients for the independent features.

Liu et al. [14] conducted a study in Yunyang, China, to evaluate the performance of multiple linear regression on village-level poverty indicators. The

model was able to identify the importance of variables that were strongly related to poverty but correlated with other variables. Xhafaj and Nurja determined factors that influence poverty using data from the Living Standards Measurement Study [15] using multiple linear regression and observed various coefficient values for different attributes.

In this work, the data was scaled using a StandardScaler function, which standardizes the features by eliminating mean and scaling features to unit variance. This was performed to make it easier for the model to analyze the coefficients when predicting the target variable.

b) Decision Tree Regressor

Decision Tree (DT) [16] is an algorithm that has a tree structure. The final leaves refer to the target or outcome variables. The features or explanatory variables are broken down into smaller subsets. Decision trees have been used both for classification and regression when handling poverty data. For predicting poverty, a regressor is used. Applications like poverty mapping and prediction have used decision tree models to make accurate estimations and even apply the same to specific regions like a study conducted in Nepal [17]. The leaves of the tree are the predicted MPI value (national/regional) and the nodes are split up into explanatory features like nutrition, child mortality, etc. A tree of depth 5 was used for predicting the MPI for all four datasets. The training set was passed through scikit-learn's decision tree model and the testing set was fitted to the model.

c) Random Forest Regressor

Akin to random forests used for classification, where the leaves are class labels, the random forest regressor [18] takes numerical values instead for the leaf nodes. Random forests are a combination of multiple prediction trees, where each tree is based on independently sampled randomized vector values. Regardless, all of the trees have the same distribution. Since multiple trees are produced in an ensemble fashion, it is known as a forest. Browne et al. [19] applied multivariate random forest prediction to estimate the correlation between malnutrition and poverty measures. The method adopted involved using two variations of random forests—independent random forests and Mahalanobis random forests on Demographic and Health Survey (DHS) data. Estimating poverty using geospatial and DHS data was performed by Zhoa et al. [20] using random forest regression. A case study was performed for Bangladesh, achieving an R2 score of 0.70 and 0.61 for Nepal.

The number of trees used for prediction was 500 and passed as the n_estimator parameter for scikit-learn's random forest regressor. The testing set was fitted to the trained model and the regressor was trained and tested accordingly.

d) XGBoost

XGBoost [21] is a scalable boosting algorithm that uses the concept of converting weak to strong learners. The algorithm was proposed for predicting a target variable given a set of explanatory variables, which makes it apt for predicting poverty using MPI data. The key concept of this algorithm is building D regression trees sequentially, where the previous trees are used to train the subsequent trees. Hence, the new tree corrects the errors of the previous trees. Li et al. analyzed data from 8040 households in Kyrgyzstan [22] and used XGBoost to profit from the model's ability to handle several independent variables. They compared the performance to a general linear model trained on the same dataset and observed the superiority in the XGBoost algorithm's performance. A study conducted by Sharma et al. [23] used XGBoost to score the poverty level of a household in a text classification context. The F1 score achieved was 0.38 for the model.

The number of trees used for prediction was 500 and passed as a parameter for the XGBoost model. The feature importance was plotted using XGBoost for the datasets, as shown in Fig. 6. For the National data in 2019, the most important feature was "Years of Schooling" whereas it was "Cooking fuel" in 2021.

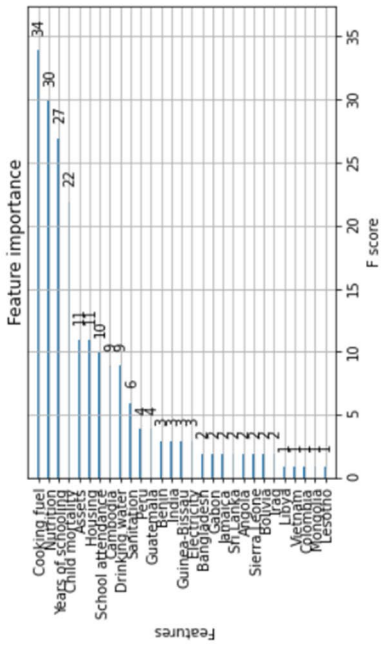
e) AdaBoost

Introduced by Robert Schapire, the Adaptive Boosting (AdaBoost) [24] algorithm is an ensemble method. A set of weak trees that are connected in series is created. Similar to the XGBoost algorithm, each subsequent tree tries to predict better than the previous. AdaBoost applies more weight instances that are harder to predict compared to those which are predicted well by the algorithm. Decision trees are used as AdaBoost's weak learners and are referred to as decision stumps. While Adaboost has been employed in poverty classification [25], the research in this paper presents a novel use-case of the AdaBoost regressor for poverty prediction on MPI data.

All of the indicators were used after pre-processing to predict MPI using AdaBoost. The number of trees used was 100 and passed as the n_estimator parameter for scikit-learn's Adaboost regressor.

f) Gradient Boosting

AdaBoost and related algorithms were further developed by Friedman [26] and called Gradient Boosting Machines, and later Gradient Tree Boosting. Gradient boosting involves a loss function that needs to be optimized. Decision trees are used as the weak learner to make predictions and the model takes an additive approach where trees are added one at a time and the existing trees are not changed. To perform the gradient descent procedure, a tree is added to the model that reduces the loss. The output of the new tree is then added to the output of the existing sequence of trees to improve the final output of the model. A policy research working paper by the World Bank [27] predicted six target variables including poverty rates, changes in poverty rates, mean welfare, etc. using



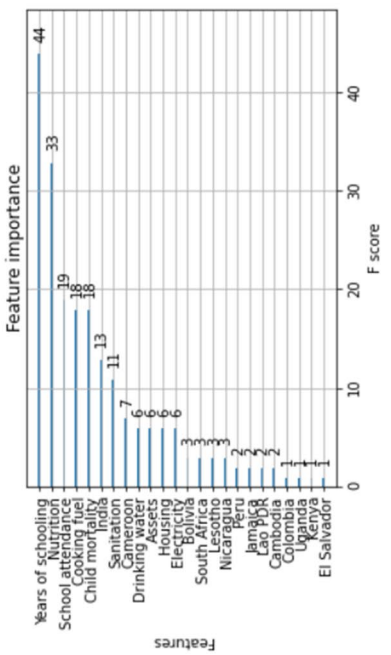


Fig. 6 Feature importance of National 2019 (left) and 2021 (right) datasets

gradient boosting. However, the model did not perform well on their dataset due to several missing input values, producing a high error value.

All of the indicators were used after pre-processing to predict MPI using Gradient Boosting. The number of trees used was 100 and passed as the N estimator parameter for scikit-learn's Gradient Boosting Regressor.

g) Linear Support Vector Regression

Vapnik proposed the Support Vector Machine algorithm in 1995 [28]. In this algorithm, independent variables are mapped to a multi-dimensional space. This is done by a hyper-plane, which is calculated optimally and splits the observations into its classes. Similar to classification, Support Vector Regressor (SVR), its regression counterpart estimates real numerical values. For example, Henrique et al. [29] used SVR for the prediction of stock prices. Bienvenido-Heurtas et al. [30] used SVR to predict fuel deprivations in Chile and achieved high correlation coefficient values of 99.5%. Similarly, SVR has been used in predicting the MPI values of various countries and regions around the globe. A linear kernel was used for both the National and Subnational datasets.

h) Ridge Regression

Ridge regression [31], commonly called L2 regression, is a regularized linear regression model that shrinks the coefficients for the input variables that are not important to the prediction task. Hence, the model complexity is reduced. In this algorithm, the dependent and independent variables are standardized for calculation. One study employed ridge regression to identify the socioeconomic factors affecting poverty using national-level data of 68 countries [32]. The algorithm detected the interrelation among the components of the multiple linear regression model used for prediction. For this work, an alpha value of 0.01 was passed as a parameter to the model.

i) Lasso Regression

Least Absolute Shrinkage Selector Operator (Lasso) regression [33] is a technique similar to ridge regression, except for one difference. Lasso regression retains or shrinks the coefficients for some features while reducing the coefficients of others to zero [34]. This is known as feature selection and is absent in the case of ridge regression. This advantage has been exploited by several researchers, including an application for selecting variables to predict poverty in Sri Lanka [35]. It was observed that lasso regression outperformed the other models used, in cases of large prediction sets. This algorithm uses L1 regularization technique. An alpha value of 0.01 was passed as a parameter to the model for this experiment.

j) ElasticNet Regression

ElasticNet is a combination of L1 and L2 regularization [36]. It has been employed to model poverty in Yogyakarta [37] where various combinations of models were combined with ElasticNet to classify poverty levels. If ElasticNet is implemented, then both Ridge and Lasso can be derived by tuning the parameters. Since there are several correlated independent variables in the dataset, the elastic net will form a group consisting of these correlated variables. If any one of the variables in this group is a strong predictor, then the entire group is included in the model building, because omitting other variables might result in losing information in terms of interpretation ability, leading to poor model performance. The model was implemented for this work using an alpha value of 0.01.

k) K-Nearest Neighbor Regression

The K-Nearest Neighbor (KNN) algorithm can be used for classification and regression problems [38]. The algorithm uses the concept of feature similarity to make predictions given new input data. This means that a value is assigned to the new input based on how closely it resembles the data in the training set. The algorithm relies on majority voting based on class membership of k-nearest samples, so the normalization of data is required to make correct predictions. The algorithm has been used to forecast economic events [39] and predict poverty using e-commerce data [40]. The latter was used to determine the level of poverty in Indonesia but there is room for improvement in terms of model accuracy.

The features in the National and Subnational datasets were scaled and transformed using MinMaxScaler before fitting the model to the dataset. The *k*-values were iterated from 1 to 30, and hyperparameter optimization was carried out through Grid Search [41]. The model was fitted to scikit-learn's GridSearchCV function, which is a cross-validation technique that will find the best value of *k* for the model. Hyperparameter optimization was used to increase the performance of the model, and GridSearchCV automates the process. Batch sizes of 5 were used for the cross-validation process.

Table 2 Algorithms used and respective parameters

Algorithm	Parameters
Multiple linear regression	Default parameters
Decision tree regressor	n_estimators = 100
Random forest regressor	n_estimators = 500
XGBoost regressor	n_estimators = 500
AdaBoost	random_state = 0, n_estimators = 100
Gradient boosting	random_state = 0, n_estimators = 100
Support vector regression	kernel = 'linear'
Ridge regression	alpha = 0.01
Lasso regression	alpha = 0.01
ElasticNet regression	alpha = 0.01
K-NN regression	n_neighbors = 30

Table 3 System Specifications of the work conducted

Aspect	Specification
CPU	Intel(R) Core(TM) i5-9300H CPU @ 2.40 GHz
GPU	Nvidia GeForce GTX 1650
Memory	8 GB @ 2400 MHz
OS	Microsoft Windows 11 Version 10.0.22000, 64-bit

Table 2 lists out the machine learning algorithms used for this work and the parameters passed to each model. The parameters were tuned to ensure maximum performance of the algorithms.

The system specifications on which the study was performed and executed are shown in Table 3. Experiments were conducted on the CPU since GPU did not provide a difference with respect to speed-up.

6 Results

In this section, the values obtained for metrics for the Machine Learning (ML) models are presented.

A) R^2 and RMSE values

The first metric used to evaluate the models is R^2 score [42]. R^2 score is defined as the percentage of variation in the dependent variable explained by variation in the independent variables, as in Eq. 2. The closer to 1 the R^2 score is, the more

Table 4 R^2 scores for algorithms

Algorithm accuracy (%)	Dataset			
	National 2019	National 2021	Subnational 2019	Subnational 2021
Multiple linear regression	0.9699	0.9668	0.9998	0.8114
Decision tree regressor	0.9626	0.9847	0.9831	0.7524
Random forest regressor	0.9887	0.9953	0.9937	0.8928
XGBoost regressor	0.9727	0.9881	0.9938	**0.9053**
AdaBoost	0.9788	0.9906	0.9826	0.7956
Gradient boosting	0.9845	0.9904	0.9959	0.8941
Support vector regression	0.9442	0.9558	0.9962	0.7944
Ridge regression	**0.9999**	**0.9999**	**0.9998**	0.8309
Lasso regression	0.9995	0.9993	0.9995	0.4424
ElasticNet regression	0.9998	0.9996	0.9996	0.4519
K-NN regression	0.9812	0.9620	0.9695	0.7521

Bold values indicate the high accuracy value

correlated the variables are, indicating validity of the regression model. Table 4 presents the R^2 scores obtained across the four datasets used.

$$R^2 = 1 - \frac{\text{RSS}}{\text{TSS}} \tag{2}$$

where RSS is the sum of squares of the residual errors and TSS is the total sum of the errors.

From Table 4, for the National 2019 dataset, Ridge Regression (L2 regularization) outperforms the other regression models with an R^2 score of 0.9999, owing to the algorithm's ability to simplify the features for prediction of MPI value. Likewise, for the National 2021 and Subnational 2019 datasets, Ridge Regression performs superior to the other algorithms. It can be observed that the other regularization techniques like Lasso and ElasticNet also perform very well on these three datasets. Applying regularization to linear regression significantly improved the performance of the model on the dataset as seen in Table 4. Variation is encountered for the Subnational 2021 dataset where it is seen that the XGBoost regressor performs the best with an R^2 score of 0.9053. This could be attributed to the XGBoost algorithm's ability to handle a large number of independent variables, as in the case of the subnational datasets. The regularization algorithms did not perform very well on the Subnational 2021 data, even after tuning the alpha values. Overall, a majority of the algorithms performed well on the four datasets.

Another important metric for the evaluation of regression algorithms is Root Mean Square Error (RMSE) which is the average of the square of the errors [43–46]. This is a way to estimate the standard deviation of the observed values from the predicted values. The higher the RMSE value, the greater the error. Equation 3 represents the RMSE calculation

Table 5 RMSE values for algorithms

Algorithm accuracy (%)	Dataset			
	National 2019	National 2021	Subnational 2019	Subnational 2021
Multiple linear regression	0.182	0.151	0.012	0.432
Decision tree regressor	0.203	0.105	0.135	0.498
Random forest regressor	0.017	0.009	0.014	0.055
XGBoost regressor	0.027	0.014	0.014	**0.052**
AdaBoost	0.024	0.012	0.024	0.076
Gradient boosting	0.021	0.012	0.012	0.054
Support vector regression	0.248	0.174	0.064	0.451
Ridge regression	**0.003**	**0.002**	**0.002**	0.083
Lasso regression	0.005	0.003	0.003	0.125
ElasticNet regression	0.003	0.003	0.003	0.132
K-NN regression	0.021	0.027	0.030	0.085

Bold values indicate the high accuracy value

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} \left(x_i - y_i\right)^2}{N}} \tag{3}$$

where N is the number of observations, x_i is the actual observation and y_i is the predicted value. Table 5 presents the RMSE values obtained for all the models across the datasets.

From Table 5, the regularization algorithms (Ridge, Lasso, ElasticNet) had the lowest RMSE values for the National 2019 and 2021 and Subnational 2019 datasets, indicating a low error rate, owing to the high R^2 scores achieved by these models. The errors of these models are significant in the Subnational 2021 dataset, but the other models like XGBoost and AdaBoost had lower RMSE values for the same. It can be observed that the boosting and ensemble methods performed well in comparison to the regularization techniques for the Subnational 2021 dataset.

Figure 7 shows a comparison of R^2 scores (%) for the National datasets, and Fig. 8 shows a comparison of R^2 scores for the Subnational datasets. The regression algorithms have performed well on the National datasets and the Subnational 2019 dataset, as shown in the bar plots. There is a significant variation in the Subnational 2021 dataset as indicated by the bars, showing that some models did not perform well.

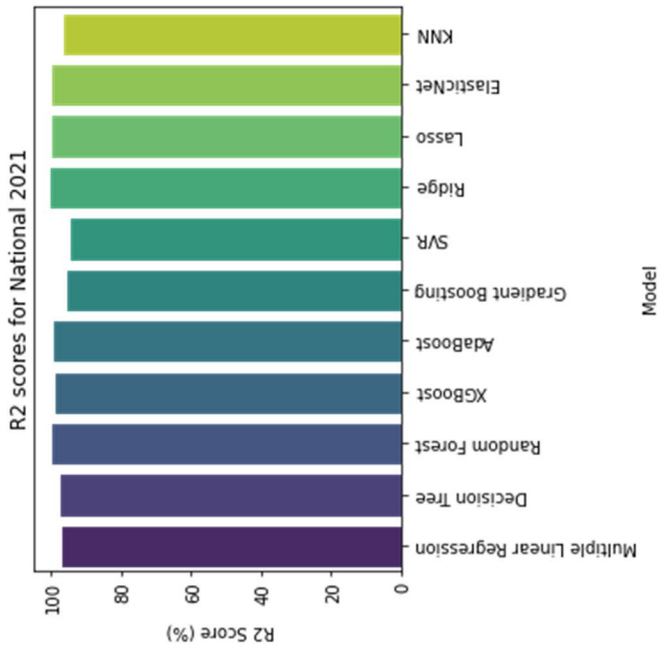
B) Prediction Error and Residuals for Regularization Models

A prediction error plot gives a view of the variance in the model by presenting the actual targets from the datasets against the predicted values generated by the model. Residual error is calculated by subtracting the predicted values $(\bar{y})$ from the observed values (y) for the target variable. The performance of the multiple linear regression model was evaluated using statistical plots which plotted the predicted against actuals and the residual error of each prediction, as shown in Fig. 9.

From Fig. 9, it can be observed that the predicted versus true graphs fit to a linear trend, indicating that the predicted and true values were similar for the multiple linear regression model. The high R^2 score is thus accounted for. The dotted line on the predicted versus residual error graphs shows the maximum error for the model on the National datasets, and it can be inferred that the error value is low for the model on both datasets.

Ridge, Lasso and ElasticNet regression were fitted to the training set to minimize loss and prevent overfitting of the model. The results of the regularization showed promising values, and prediction error and residual graphs were visualized for all of the datasets.

From the plots for the three models in Fig. 10, the errors for the National 2019 dataset are shown. It can be seen that the Ridge and ElasticNet models have less errors than the Lasso model. Overall, the regularization techniques improve the accuracy of the linear regression model, which aligns with the expected outcome.



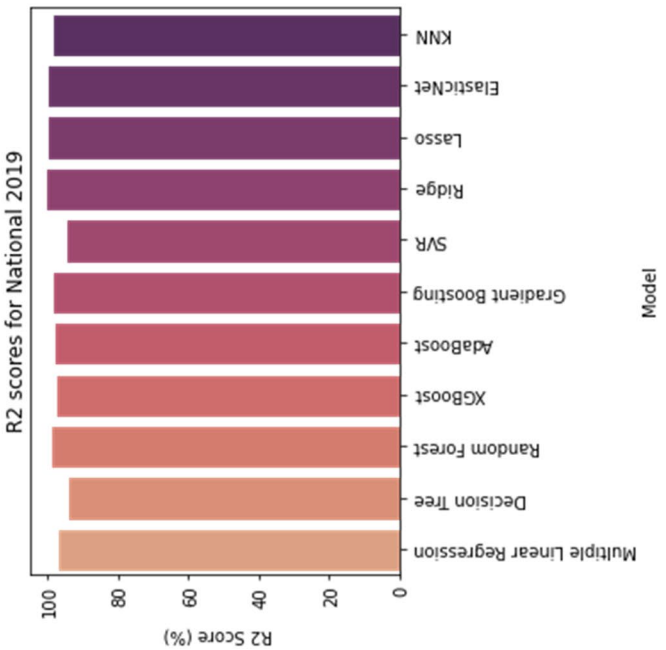
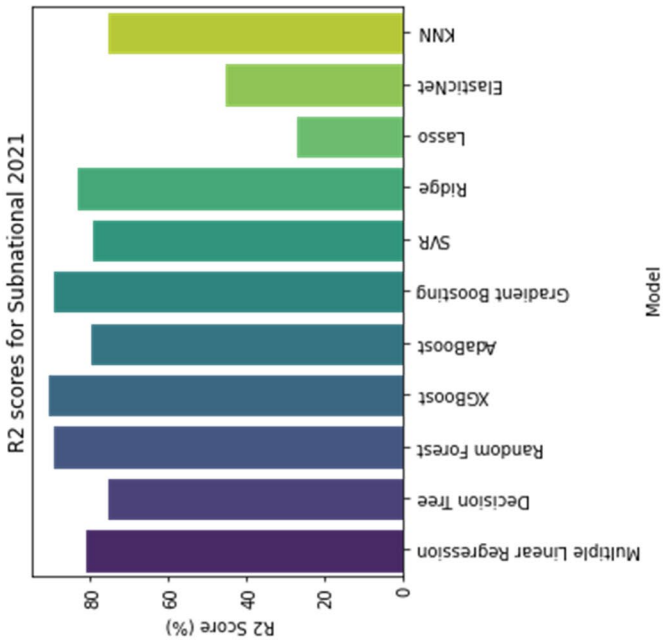


Fig. 7 Comparison of R^2 scores for National datasets



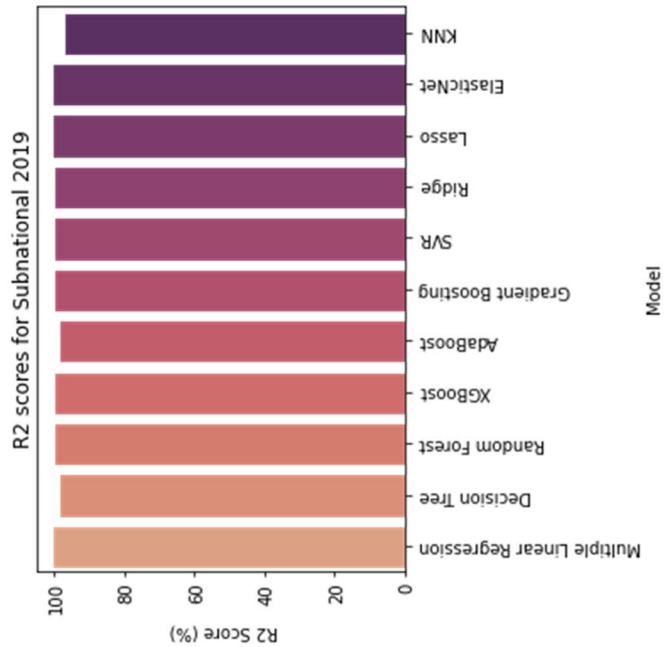
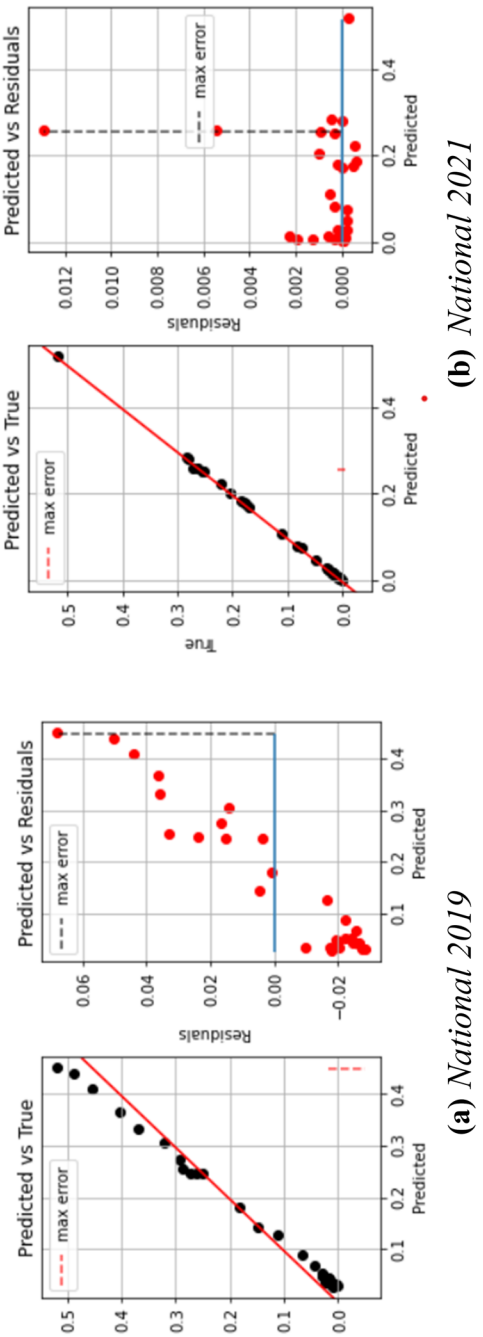


Fig. 8 Comparison of R^2 values for Subnational datasets



(a) *National 2019* **(b)** *National 2021*

Fig. 9 Predicted vs Actuals and Residuals for National Datasets

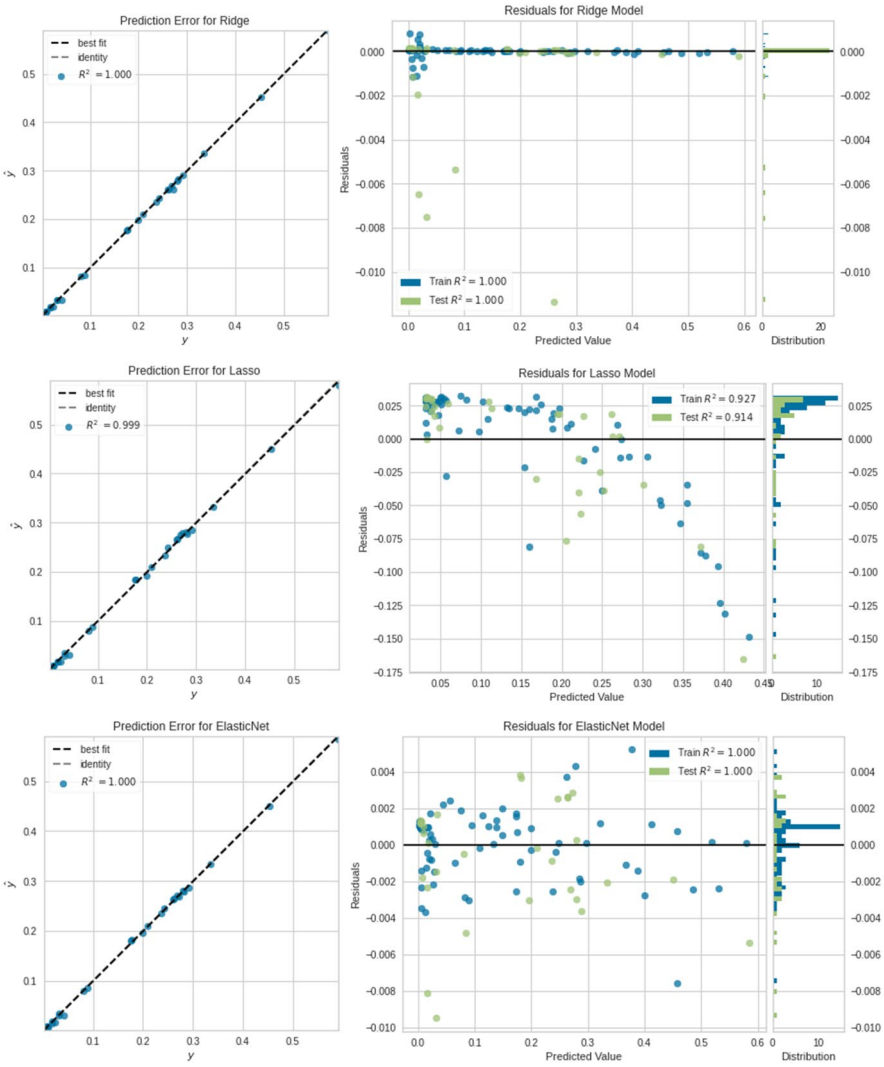


Fig. 10 Prediction Error and Residual plots for Regularized regression models (National 2019)

In Fig. 11, a comparison of the prediction error and residuals for three models fitted on the National 2021 dataset can be seen. The Ridge and ElasticNet model [47–51] shows a slightly higher values for R2 scores compared to the Lasso model, similar to the performance of the models on the National 2019 dataset. It can be inferred that the Ridge regression has less residual errors than ElasticNet from the way the points are more evenly distributed in the residual error plot, whereas ElasticNet shows more variance in the data points.

The Subnational 2019 dataset featured more independent variables than the National datasets. As a consequence, it was important to use regularization on

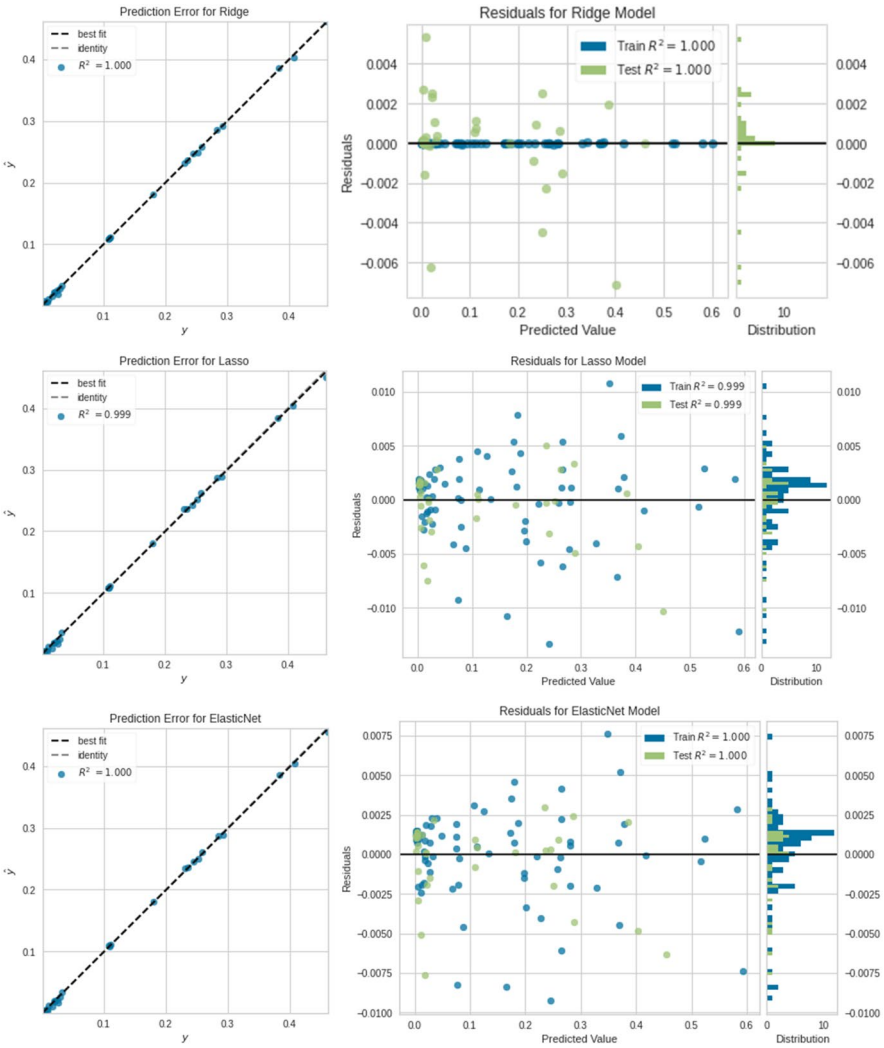


Fig. 11 Prediction Error and Residual plots for Regularized regression models (National 2021)

the linear regression model fitted to the dataset. From Fig. 12, it can be inferred that all three regularized models had low to close to no residual errors, but the difference in their performance is attributed to the difference in the variation and distribution of the points.

Akin the Subnational 2019 dataset, the predicted and residual errors for the Subnational 2021 dataset can be observed in Fig. 13. In comparison to the plots generated for the models on the other three datasets, the plots in Fig. 13 show the greatest amount of variance. This variance leads to a higher error rate and lower performance of the regularization techniques on the Subnational 2021 data.

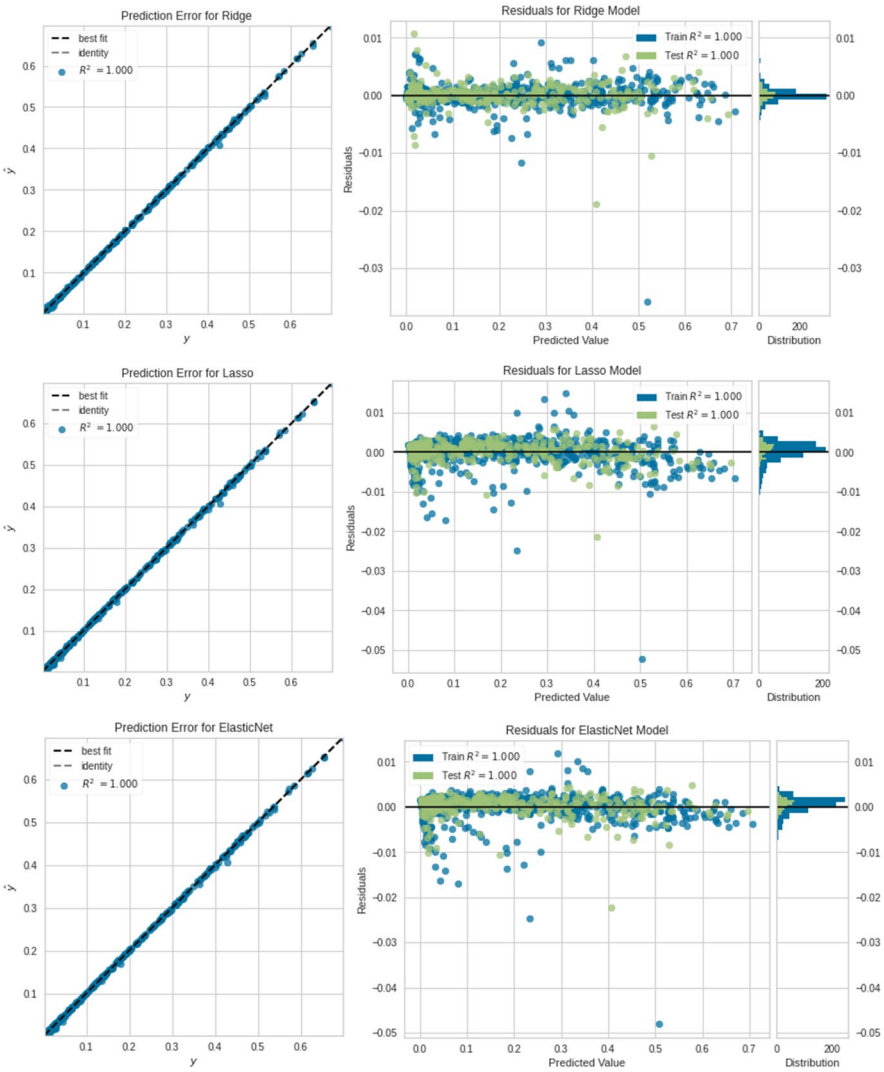


Fig. 12 Prediction Error and Residual plots for Regularized regression models (Subnational 2019)

While the Ridge model shows a fair R^2 score, Lasso and ElasticNet regression performed poorly on this dataset.

C) Optimal Values of *K*

For the KNN algorithm, it is important to determine the optimal value of k, which defines how many clusters the data should be divided into. Determining this value reduces the effect of noise and the elbow method was employed to select the optimal

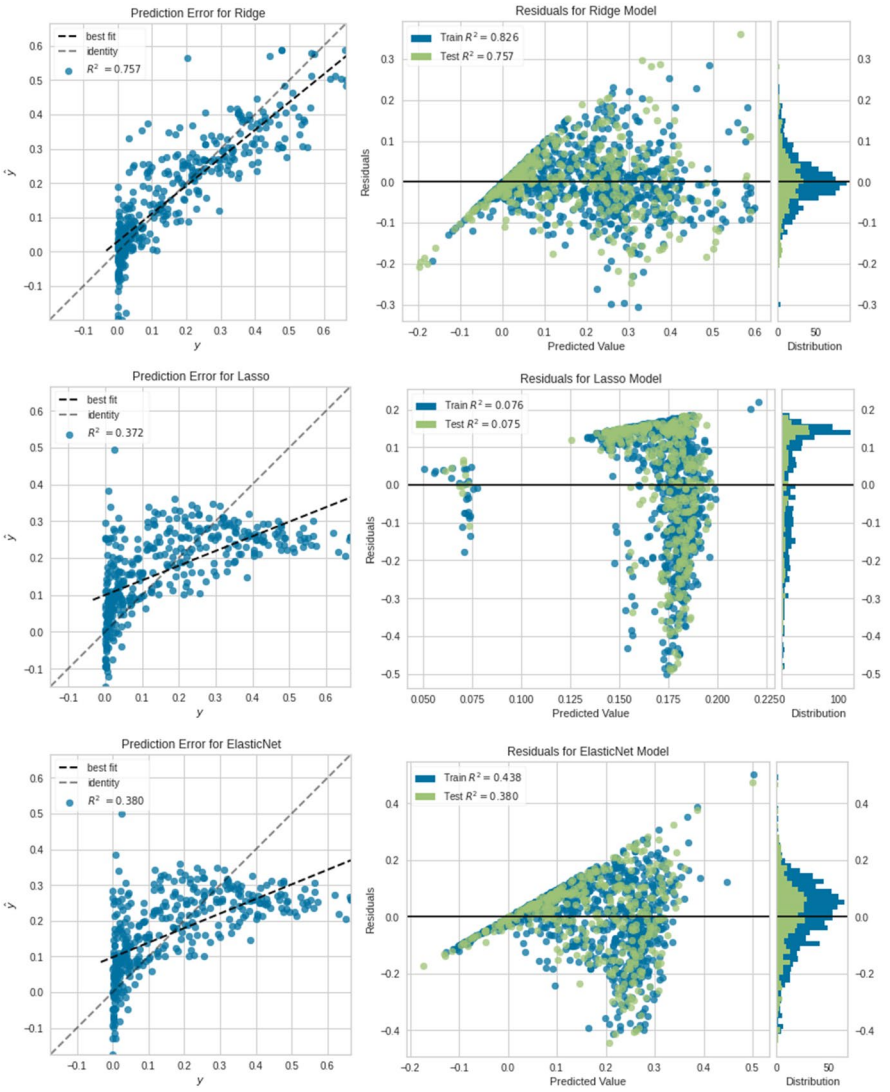
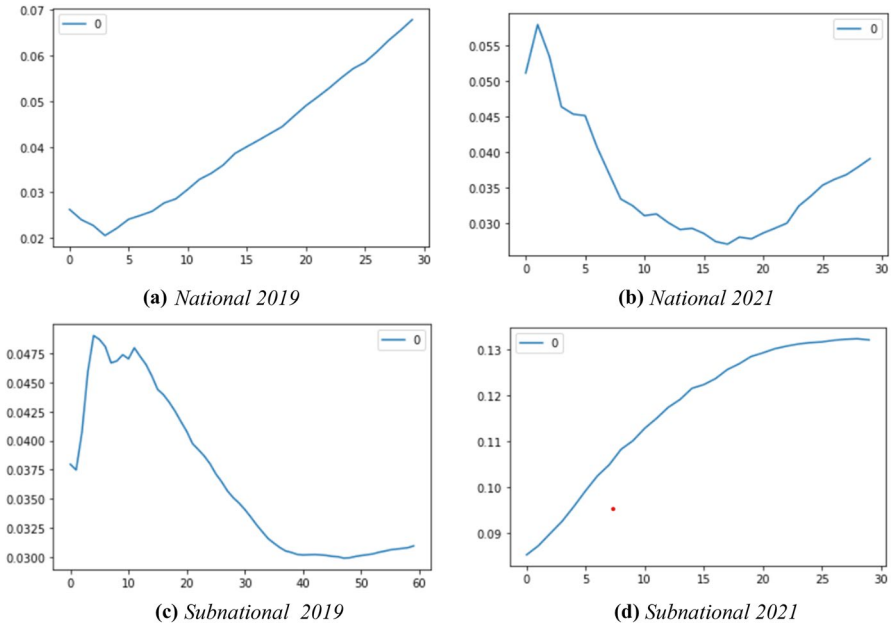


Fig. 13 Prediction Error and Residual plots for Regularized regression models (Subnational 2021)

number of clusters. Figure 14 consists of the RMSE elbow curves for the KNN algorithm applied on the four datasets.

Table 6 shows the optimal *k*-values obtained for each dataset used.

D) Tree-Based Regression Analysis



(a) *National 2019* **(b)** *National 2021*

(c) *Subnational 2019* **(d)** *Subnational 2021*

Fig. 14 Elbow Curves for RMSE values for values of k

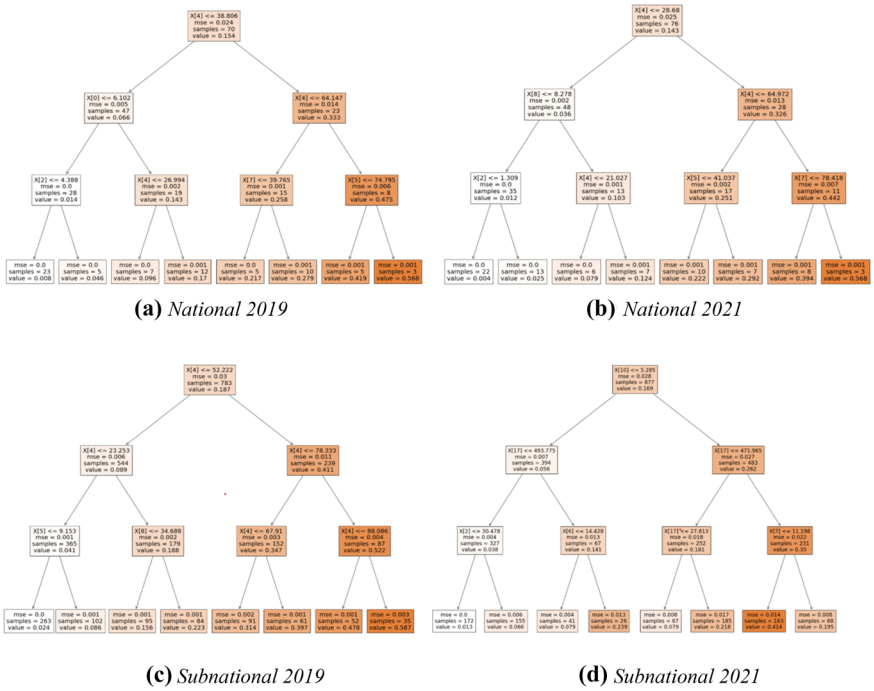
The decision tree algorithm performed consistently across all the datasets, with a slight drop in performance for the Subnational 2021 dataset. Figure 15 shows the tree representation with the mean square error (MSE) values obtained for different features in the testing set. A depth of 3 was used for ease of viewing the diagram; the actual depth used during training was 5 for the decision tree. As seen in the diagram, the MSE values in the leaf nodes of the trees are low, indicating that the model has performed very well on the dataset.

Figure 15c and d depicts the decision tree results for the Subnational datasets (2019 and 2021 respectively). The leaf nodes of Fig. 15c show low MSE values when compared to the values in Fig. 15d. The Subnational 2021 dataset shows relatively higher error rates in comparison to the other datasets, which indicates a drop in performance. The parameters of the tree were tuned to increase the R2 score, which proved to be efficacious.

While viewing the MSE values in the form of a decision tree can be effective, Fig. 16 depicts regression trees for the Subnational datasets, providing a close-up view of the features that influence the training of the model to predict regional MPI values. The vertical dashed lines indicate the split point in the feature space and the black wedge indicates the exact split value. The target prediction is indicated by the leaf nodes. From Fig. 16, it is easy to identify the difference between the training of the datasets. The Subnational 2019 dataset shows a positive correlation between the training points, whereas the Subnational 2021 dataset shows a more scattered correlation, which explains the drop in performance of the model on the same.

Table 6 Optimal *K*-values for all datasets obtained through Grid Search CV

Dataset	Optimal *K*-value through grid search CV
National 2019	4
National 2021	4
Subnational 2019	28
Subnational 2021	2



(a) *National 2019* **(b)** *National 2021*

(c) *Subnational 2019* **(d)** *Subnational 2021*

Fig. 15 Decision tree MSE results (depth = 3 for ease of viewing)

7 Conclusion

Currently, the first sustainable development goal of the United Nations, predicting poverty is a difficult task that requires sufficient data and research. Harnessing machine learning to predict poverty is one way of approaching the issue. Measuring the extent to which factors included in the MPI affect the overall picture of poverty is a crucial step in decision-making related to poverty and allows organizations to focus on areas in need. Using MPI data instead of household surveys can facilitate a clearer picture of exactly which indicators are lacking in specific regions.

The research presented applies multiple regression models for predicting MPI on a national and subnational basis across pre-COVID data (2019 MPI) and data during

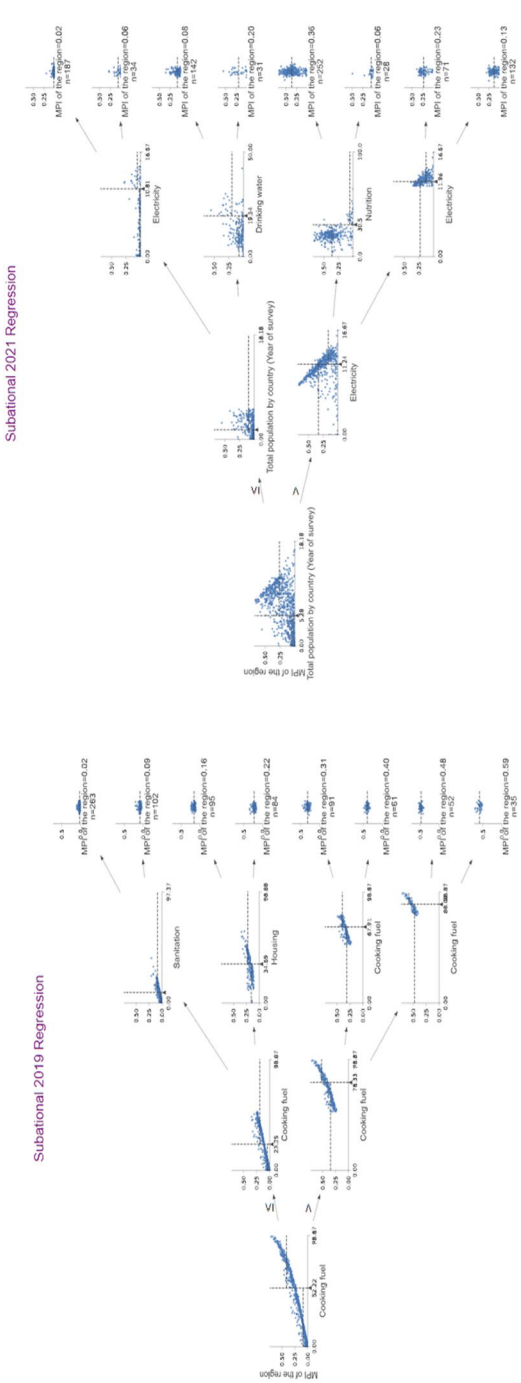


Fig. 16 Regression tree for Subnational Datasets (left: 2019, right: 2021)

COVID-19 (2021 MPI). From the models used, it can be concluded that the ridge regression model worked the best across the National datasets and the Subnational 2019 dataset, with an alpha value of 0.01. The XGBoost regressor performed the best on the subnational 2021 dataset. The other regression models performed well on the datasets. Predicted and residual error plots depict the variance in data points and explain the performance of the regularized models on the datasets, and other regression graphs paint a picture of the predictions made by the models. Cross-validation was used to prevent overfitting or under-fitting to the models. Through feature selection and analyzing feature importance, we can conclude that the most important determinants of poverty are nutrition, cooking fuel, years of schooling, school attendance and child mortality, according to the *F*-scores obtained. The importance of these features has changed during the pandemic, indicating a shift in the way they affect poverty.

However, there are limitations to the current work. First, the MPI data can be more diverse and detailed to further improve the accuracy and enable a deeper explanation of poverty determinants. Second, some of the models did not achieve optimal results on the datasets. Higher R^2 scores can be obtained by tuning the parameters further and improving the algorithms. Future scope includes finding out how closely interrelated the MPI data is across the years and how to further leverage this data to map poverty effectively.

Data availability The data associated with this publication is available with the authors.

References

1. World Bank Group, Poverty and Shared Prosperity 2020, Accessed: 26 Feb 2022
2. Banks, L.M., Kuper, H., Polack, S.: Poverty and disability in low- and middle-income countries: a systematic review. PLoS One **13**(9), e0204881 (2018)
3. Zixi, H.: Poverty Prediction Through Machine Learning, 2021 2nd International Conference on E-Commerce and Internet Technology (ECIT) (2021), pp. 314–324
4. Jean, N., Burke, M., Xie, M., Davis, W.M., Lobell, D.B., Ermon, S.: Combining satellite imagery and machine learning to predict poverty. Science **353**(6301), 790–794 (2016)
5. Alkire, S., Nogalesa, R., NaïriQuinn, N., Suppaade, N.: Global multidimensional poverty and COVID-19: a decade of progress at risk? Soc. Sci. Med. **291**, 114457 (2021)
6. Alkire, S., Kovesdi, F., Pinilla-Roncancio, M., Scharlin-Pettee, S.: Changes over time in the global multidimensional poverty index and other measures: towards national poverty reports, OPHI Research in Progress 57a, Oxford Poverty and Human Development Initiative, University of Oxford (2020d)
7. Anderson, R.M., Heesterbeek, H., Klinkenberg, D., Hollingsworth, T.D.: How will country-based mitigation measures influence the course of the COVID-19 epidemic? Lancet **395**(10228), 931–934 (2020)
8. Tavares, F.F., Betti, G.: The pandemic of poverty, vulnerability, and COVID-19: evidence from a fuzzy multidimensional analysis of deprivations in Brazil. World Dev. **139**, 105307 (2021)
9. Huanga, Y., Jiao, W., Wang, K., Li, E., Yan, Y., Chen, J., Guo, X.: Examining the multidimensional energy poverty trap and its determinants: an empirical analysis at household and community levels in six provinces of China. Energy Policy **169**, 113193 (2022)
10. Alkire, S., Santos, M.E.: Multidimensional poverty index, Oxford Poverty & Human Development Initiative (OPHI) (2010)

11. Wolff, E.N.: Wealth distribution, international encyclopedia of the social & behavioral sciences (2nd Edn), (2015), pp. 450–455. https://doi.org/10.1016/B978-0-08-097086-8.71017-8
12. Groß, J.: Linear regression, lecture notes in statistics 175, (2003) https://doi.org/10.1007/978-3-642-55864-1
13. Uyanık, G.K., Güler, N.: A study on multiple linear regression analysis. Procedia. Soc. Behav. Sci. **106**, 234–240 (2013)
14. Liu, M., Hu, S., Ge, Y., Heuvelink, G.B.M., Ren, Z., Huang, X.: Using multiple linear regression and random forests to identify spatial poverty determinants in rural China. Spatial. Statistics. (2021). https://doi.org/10.1016/j.spasta.2020.100461
15. Xhafaj, E., Nurja, I.: Determination of key factors that influence poverty through econometric models. Eur. Sci. J. **10**(24), 65–72 (2014)
16. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**, 81–106 (1986)
17. Bilton, P.A.: Tree-based models for poverty estimation. https://mro.massey.ac.nz/handle/10179/11218. (2016) Accessed: 15 Jan 2021
18. Breiman, L.: Random forests. Mach. Learn. **45**, 5–32 (2001)
19. Browne, C., Matteson, D.S., McBride, L., Hu, L., Liu, Y., Sun, Y., Wen, J., Barrett, C.B.: Multivariate random forest prediction of poverty and malnutrition prevalence. PLoS One (2021). https://doi.org/10.1371/journal.pone.0255519
20. Zhao, X., Yu, B., Liu, Y., Chen, Z., Li, Q., Wang, C., Wu, J.: Estimation of poverty using random forest regression with multi-source data: a case study in Bangladesh. Remote. Sens. **11**, 375 (2019)
21. Chen, T., Guestrin, C.: XGBoost: A scalable tree boosting system, In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016), pp. 785–794
22. Li, Q., Yu, S., Échevin, D., Fan, M.: Is poverty predictable with machine learning? A study of DHS data from Kyrgyzstan. Socio-Economic. Plan.. Sci. **81**, 101195 (2021)
23. Sharma, A., Rathod, J., Pol, R., Gajbhiye, S.: Poverty prediction using machine learning. Int. J. Computer Sci. Eng. **7**(3), 946–949 (2019)
24. Schapire, R.E., Robert, E.: Explaining AdaBoost. Empirical. Inference. (2013). https://doi.org/10.1007/978-3-642-41136-6_5
25. Alsharkawi, A., Al-Fetyani, M., Dawas, M., Saadeh, H., Alyaman, M.: Poverty classification using machine learning: the case of jordan. Sustainability **13**, 1412 (2021)
26. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. Stat. **29**(5), 1189–1232 (1999)
27. Aguilar, R.A.C., Mahler, D.G., Newhouse, D.: Nowcasting Global Poverty, Policy Research Working Paper 9860, World Bank (2021)
28. Cristianini, N., Ricci, E.: Support vector machines, encyclopedia of algorithms, Springer (2008)
29. Henrique, B.M., Sobreiro, V.A., Kimura, H.: Stock price prediction using support vector regression on daily and up to the minute prices. J. Finance. Data. Sci. **4**(3), 183–201 (2018)
30. Bienvenido-Heurtas, D., Pulido-Arcas, J.A., Rubio-Bellido, C., Perez-Fargallo, A.: Prediction of fuel poverty potential risk index using six regression algorithms: a case-study of chilean social dwellings. Sustainability **13**, 2426 (2021)
31. Hoerl, A.E., Kennard, R.W., Baldwin, K.F.: Ridge regression—some simulations. Commun. Stat. **4**, 105–123 (1975)
32. Sufian, A.J.M.: An analysis of poverty—a ridge regression approach, IMSCI 2010—4th International Multi-Conference on Society, Cybernetics and Informatics, Proceedings 2 (2010), pp. 118–123
33. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. Royal. Statistical Soc. Series B. (Methodological) **58**(1), 267–288 (1996)
34. Tibshirani, R.: Regression shrinkage and selection via the lasso: a retrospective, J. Royal. Statistical. Soc., Series B. (Methodological). 73(3) (2011), pp. 273–282, https://webdoc.agsci.colostate.edu/koontz/arec-econ535/papers/Tibshirani(JRSS-B2011).pdf
35. Afzal, M., Hersh, J., Newhouse, D.: Building a better model: variable selection to predict poverty in Pakistan and Sri Lanka, World Bank Research Working Paper (2015)
36. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. J. Royal. Statistical. Soc. Series B. (Statistical Methodol). **67**(2), 301–320 (2005)
37. Sihombing, P.: Regularized ordinal regression with elastic net approach (case study: poverty modeling in Yogyakarta Province 2018). CAUCHY **6**, 296–304 (2021)

38. Guo, G., Wang, H., Bell, D., Bi, Y.: KNN model-based approach in classification, on the move to meaningful internet systems 2003: CoopIS, DOA, and ODBASE. OTM (2003). https://doi.org/10.1007/978-3-540-39964-3_62
39. Imandoust, S.B., Bolandraftar, M.: Application of K-nearest neighbor (KNN) approach for predicting economic events: theoretical background. Int. J. Eng. Res. Appl. **3**(5), 605–610 (2013)
40. Aulia T.F., Wijaya, D.R., Hernawati E., Hidayat, W.: Poverty level prediction based on E-commerce data using k-nearest neighbor and information-theoretical-based feature selection, 2020 3rd International Conference on Information and Communications Technology (ICOIACT) (2020), pp. 28–33, https://doi.org/10.1109/ICOIACT50329.2020.9332083
41. Liashchynskyi, P.B.: Grid search, random search, genetic algorithm: a big comparison for NAS, https://arxiv.org/abs/quant-ph/1912.06059 (2019)
42. Figueiredo, D., Júnior, S., Rocha, E.: What is R2 all about?. Leviathan-Cadernos de Pesquisa Polútica **3**, 60–68, (2011). https://doi.org/10.11606/issn.2237-4485.lev.2011.132282
43. Botchkarev, A.: Performance metrics (error measures) in machine learning regression, forecasting and prognostics: properties and typology, arXiv preprint. https://arxiv.org/abs/quant-ph/1809.03006 (2018)
44. Satapathy S.K., Dehuri, S., Jagadev, A.K., Mishra, S.: EEG brain signal classification for epileptic seizure disorder detection, Elsevier Publication, 1st Eds, ISBN- 9780128174265, Feb 2019
45. Satapathy, S.K., Dehuri, S., Jagadev, A.K.: Weighted majority voting based ensemble of classifiers using different machine learning techniques for classification of EEG signal to detect epileptic seizure. Informatica **41**, 99–110 (2017)
46. Satapathy, S.K., Jagadev, A.K., Dehuri, S.: An empirical analysis of training algorithms of neural networks: a case study of EEG signal classification using java framework. In: Jain, L.C. et al. (eds.), vol 309, Advances in intelligent systems and computing. Springer, pp 151–160, (2015)
47. Sah, S., Dhanalakshmi, S.R., Mohanty, S.N., Alenezi, F., Polat, K.: Forecasting COVID-19 pandemic using prophet, ARIMA, and hybrid stacked LSTM-GRU Models in India. Computational. Math. Methods. Med. (2022)
48. Shome, D., Kar, T., Mohanty, S.N., Tiwari, P., Muhammad, K., AlTameem, A., Zhang, Y., Saudagar, A.K.J.: COVID-transformer: interpretable COVID-19 detection using vision transformer for healthcare. Int J Env Res Public Health **18**(21), 1–14 (2021)
49. Mangla, M., Sharma, N., Mohanty, A., Satpathy, S., Mohanty, S.N., Choudhury, T.: Geospatial multivariate analysis of COVID-19: a global perspective. Geo J. (2021)
50. Shankar, K., Mohanty, S.N., Yadav, K., Gopalakrishnan, T.: Automated COVID-19 diagnosis and classification using convolutional neural network with fusion based feature extraction model. Cogn. Neurodyn. **16**(1), 22–34 (2021)
51. Dash, S., Chakravati, S., Mohanty, S.N., Patnaik, C.R., Jain, S.: A deep learning method to forecast COVID-19 outbreak. N. Gener. Comput. **39**(2), 437–461 (2021)

Authors and Affiliations

Sandeep Kumar Satapathy[1] · Shreyaa Saravanan[1] · Shruti Mishra[1] · Sachi Nandan Mohanty[2]

Sandeep Kumar Satapathy
sandeepkumar04@gmail.com

Shreyaa Saravanan
shreyaa.saravanan2018@vitstudent.ac.in

Sachi Nandan Mohanty
sachinandan09@gmail.com

[1] School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, Vandalur-Kelambakkam Road, Chennai, Tamil Nadu 600127, India

[2] School of Computer Science and Engineering (SCOPE), VIT-AP University, Amaravati, Andhra Pradesh, India