Utilizing Geotagged Data for Estimating Maternal and Child Health Indicators using Machine Learning

IIT Internship Program

Debajyoti Dasgupta
Department of Computer Science
IIT Kharagpur
debajyotidasgupta6@gmail.com

Rushil Venkateswar
Department of Computer Science
IIT Kharagpur
rushilv14@gmail.com

Piran Karkaria
Industrial and Systems Engineering
IIT Kharagpur
pirankarkaria@gmail.com

Abstract

In this research, we delve into the fusion of machine learning and geospatial data to estimate pivotal indicators of maternal and child health (MCH) within low- and middle-income countries (LMICs). Capitalizing on satellite imagery combined with geotagged datasets, we endeavor to provide precise estimations of crucial indicators, encompassing child undernutrition, anemia, mortality rates, and episodes of childhood illnesses, right down to the granularity of village and neighborhood delineations. Through rigorous model comparisons using the mean column-wise root mean squared error (MCRMSE) as our benchmark, the Random Forest model initially emerged as the frontrunner with an MCRMSE score of 11.11011 post hyperparameter fine-tuning. However, further advancements revealed the superior efficacy of ensemble models. Our innovative Boosting on Errors methodology significantly outperformed the former, attaining an MCRMSE score of 10.759—a commendable improvement of 3.2%. This evolution in performance underscores the potential that ensemble techniques harbor in this domain. These breakthroughs possess profound ramifications for the orchestration of targeted public health strategies within LMICs. By leveraging the potency of machine learning paired with geospatial insights, we present an avant-garde methodology to transcend the constraints of conventional surveys, paving the way for real-time, insightful assessments of MCH dynamics. Future avenues of research are encouraged to refine and extrapolate these modeling techniques across a broader spectrum of LMIC contexts.

1 Introduction

Reliable civil registration and vital statistics systems are integral to effective health systems and policy-making [1]. However, in many low- and middle-income countries (LMICs), such systems remain underdeveloped or non-existent [2]. Consequently, maternal and child health (MCH) indicators, which are crucial to monitoring progress towards global health goals, often rely on data from expensive nationally representative household surveys [3]. While valuable, these surveys present significant limitations: they typically sample less than 2% of a country's communities and the data obtained can quickly become outdated [4].

This research project seeks to harness the power of machine learning, satellite imagery, and geotagged data to provide a potential solution to these issues. We hypothesize that child undernutrition indicators, anemia in children and women of reproductive age, child and maternal mortality, and childhood

illness episodes can be accurately estimated at the village and neighborhood level using these data sources [5]. If successful, this approach would allow for real-time monitoring of MCH indicators and their changes over time, a capability that is unattainable with current methodologies [6].

For this study, we utilized a dataset generated from the Demographic and Health Surveys (DHS) of 59 countries in combination with various sources of satellite images [7]. Our project achieved impressive results by employing machine learning techniques, with the Random Forest model outperforming others in terms of the Mean Column-wise Root Mean Squared Error (MCRMSE) [8]. The outcomes of this research could pave the way towards more timely, accurate, and comprehensive monitoring of health indicators in LMICs, helping to inform policy and intervention design to improve MCH outcomes.

The dataset consists of around ten thousand features derived from satellite images, each row representing an aggregated administrative region. The task is multi-output regression with 6 target variables. Historically, random forest regressor has been adopted as the de facto model for bigger datasets [9], and we have adopted it as well. It is excellent at finding feature importance, and we observe that the features picked by the model trained on the total dataset, in turn, give rise to better and more efficient models for the task. We were able to achieve an MCRMSE of 10.75958 which is at the current point much higher in score as compared to the other challengers in the competition.

2 Related Work

Several previous studies have employed satellite data in efforts to predict and track health indicators, particularly in developing regions where collecting such data can be challenging and expensive.

Numerous teams have collated datasets with the objective of establishing a correlation between publicly accessible data and the results from DHS surveys. One such dataset is **SustainBench**, a collection of benchmarks dedicated to sustainability [10]. SustainBench has compiled a dataset comprising: a) satellite imagery from NASA's LandSat and b) street-level imagery from Mapillary, covering 56 nations. This dataset houses roughly 100,000 training images along with six DHS indices: asset wealth index, child mortality rate, women's BMI, women's education, water index, and sanitation index. Regarding the health indices, the dataset possesses 94,866 BMI labels calculated from 1,781,403 non-pregnant women of reproductive age (15-49). Additionally, it includes 105,582 labels pertaining to child mortality rates, derived from the data of 1,936,904 children below the age of 5.

In the study by Irvin et al. [11], they attempted to automate this data collection process by using remote sensing coupled with deep learning. Their research focused on using Convolutional Neural Networks (CNNs) to predict poverty and malnutrition directly from satellite imagery. While they found that predicting malnutrition proved to be challenging, they were successful in classifying impoverished regions with relatively high accuracy.

In a separate study, Reddy Nangi and Tantivasadakarn [12] focused on predicting Global Health Indicators to assess food security and societal health. They hypothesized that statistical machine learning models could benefit from knowledge sharing to improve their performance in predicting these health indicators. Therefore, they trained Deep Learning models combined with Multi-Task learning techniques to predict these indicators from satellite data. They built regression models specifically to predict Women's Body Mass Index (BMI) and Child Mortality Rate under 5 (CMR) using satellite image feature data. Their work showed that multi-task learning improved the performance for the CMR model, but did not significantly enhance the BMI task due to task difficulty and missing data.

Another study by Gargeya [13] aimed to correlate publicly available data sources with Demographic and Health Surveys (DHS) data from prior years. The research proposed that a system capable of computationally predicting DHS health indicators could assist in better resource allocation and make global health monitoring faster and less expensive.

Lastly, Khosla et al. [14] utilized NASA's Landsat database satellite images to predict malnourishment and child mortality rates. They evaluated the performance of various computer vision models and metadata-enhanced fusion models on ordinal discretizations of the outcome variables. Their best model, a fine-tuned Vision Transformer pre-trained on ImageNet, achieved notable results in improving over the random baseline. This research provides suggestive evidence that the model

attends to sociologically relevant aspects of the images and indicates future work should entail further enhancements of this model and extensions to other measures capturing a region's health outcomes.

3 Approach

Figure 1 gives a high-level overview of the flow of the approach that will be described in a step-by-step fashion in the following section, along with the difficulties and mitigations planned.

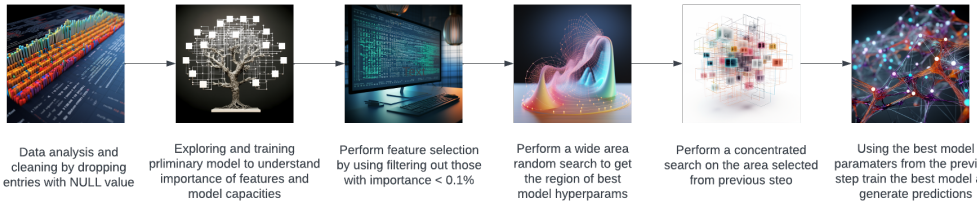

Figure 1: High-Level overview of the flow of approach

3.1 Handling Missing Data

Our first encounter in estimating key indicators of maternal and child health (MCH) status from satellite imagery and geotagged data was the high prevalence of missing data within the dataset. Imputation is a commonly used technique for dealing with such issues [15]. However, considering the volume and complexity of the missing data in our case, the imputation process posed significant challenges, and we are currently still exploring various imputation methods for our dataset.

Given these circumstances, we opted for an alternative approach to handling the missing data in the initial stages of our project. We proceeded with data cleaning, which involved dropping any columns containing at least one Null value. The primary intuition behind this is that most of the rows which contain null values are more than 50% empty and are not providing useful information even after imputing. Despite its severity, this operation left us with a considerable number of features (mode than 11000) for the subsequent phases of our study. Data cleaning is a crucial step in any machine learning project as it helps to improve the quality and reliability of the model results [16].

3.2 Exploring Machine Learning Models

The next stage in our approach involved exploring various machine learning model architectures. We specifically selected models supported by Python packages and capable of managing large datasets. Our selection included gradient boosting models such as XGBoost [17] and LightGBM [18], neural network-based models like TabNet [19] and Pytorch-based Deep Neural Networks [20], and ensemble methods like Random Forests [21]. Of all these models, the Random Forest model consistently exhibited superior performance across various evaluation metrics. In addition to individual models, we delved into ensemble methods to potentially boost our model's performance. Specifically, we tried methods such as stacking [22], voting ensemble [23], and Boosting on Errors [24]. Among these ensemble strategies, Boosting on Errors emerged as the top-performing technique, as will be illustrated in subsequent sections of this paper.

3.3 Feature Selection

Subsequent to the model selection, we embarked on feature selection using a lightweight model. The primary goal was filtering out less relevant features from the vast volume of feature data, simplifying the model, and potentially enhancing its predictive performance. This selection was based on feature importance, a measure of the contribution of each feature to the model's predictive power [25]. To concretize our feature selection approach, we fitted a Random Forest model over the entire dataset. Once trained, the model provided a ranked list of features based on their importance. While numerous features contributed to the model's prediction, a clear demarcation was observed after the top 100 features. Beyond this point, the feature importance values dropped significantly, often falling below

0.01. Given the negligible contribution of these subsequent features, we decided to retain only the top 100 features for our final modeling. This decision was grounded in the premise that features with importance values below 0.01 often add more noise than value, thus not significantly enhancing the model's performance.

3.4 Boosting on Error Ensemble

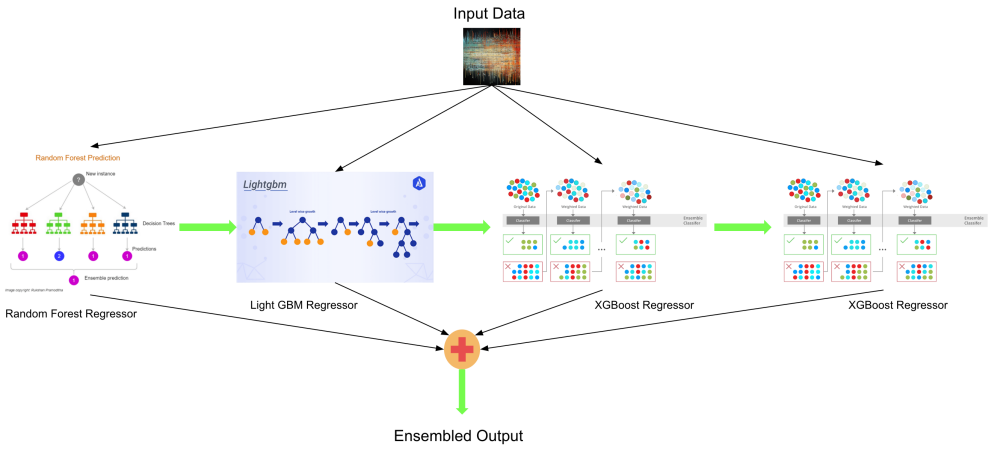

Figure 2: Final Boosting on Error model that performs the best

Our *Boosting on Error* strategy is encapsulated within the `BoostOnErrorEnsemble` class. This class, inheriting from `BaseEstimator` and `TransformerMixin`, orchestrates a cascading error refinement mechanism.

- **Initialization**: The constructor initializes four primary models.
  - A Random Forest Regressor serves as the foundational model (Level 1).
  - For Level 2, a dictionary of LightGBM Regressors is instantiated, with a dedicated model for each predictive label.
  - Level 3 and Level 4 employ XGBoost Regressors to further refine prediction errors.
- **Fitting**: During the fit phase, the class follows a structured methodology:
  1. The base model (Random Forest) is trained on the input data, `X`, and the target labels, `y`.
  2. The residuals, or error (`e_1`), between the actual values and predictions from the base model are computed.
  3. Each of the LightGBM models in Level 2 is trained on the original features of `X` but with the residuals (`e_1`) as the target. The predictions from these models are then used to further refine the residuals, yielding `e_2`.
  4. The Level 3 XGBoost model is trained using `X` and the updated residuals `e_2`. Subsequent residuals, `e_3`, are computed by subtracting the predictions of the Level 3 model from `e_2`.
  5. Lastly, the Level 4 XGBoost model is trained using `X` and the residuals `e_3`.
- **Prediction**: During prediction, the ensemble aggregates the results from all models. The final prediction is a summation of predictions from the Random Forest (Level 1), the collection of LightGBM models (Level 2), and both XGBoost models from Levels 3 and 4.

The orchestrated synergy of these models, built atop the residuals of their predecessors, exemplifies the essence of our *Boosting on Error* approach. This methodology seeks to harness the strengths of diverse models and progressively refines predictions by focusing on the errors of preceding models. This model can be visualize in the Fig. 2.

3.5 Hyperparameter Tuning

Having determined the most relevant features, we tuned the model hyperparameters. This step is indispensable in machine learning as it assists in optimizing the model's performance by searching for the most suitable combination of hyperparameters [26].

Our hyperparameter tuning process was bifurcated into two distinct stages. Initially, we conducted a random search across an expansive range of hyperparameters. This preliminary phase enabled us to identify a potentially favorable area within the hyperparameter space. Subsequently, we employed Grid Search within this identified zone to meticulously refine our selection of hyperparameters. This two-pronged approach ensures a comprehensive, yet efficient hunt for the most optimal hyperparameters.

Furthermore, this Grid Search was not confined solely to our primary models. We expanded its application to our ensemble techniques, specifically the stacking and voting ensemble, to ascertain the best-performing weights for amalgamating the models. This meticulous approach is pivotal to harmoniously unite multiple models and to achieve enhanced performance.

For our Boosting on Error strategy, each model was fine-tuned individually. First, a Random Forest model served as our level 1 base model. LightGBM was introduced at level 2, comprising six separate models, each tailored for one of the six predictive labels, to boost the base model's predictions. This boosting technique rectifies the prediction errors stemming from the level 1 model. At levels 3 and 4, XGBoost models were introduced, each refining the predictions further by reducing the error residues of the preceding models. This layered approach ensures a progressive refinement in predictions, with each subsequent level attempting to minimize the errors from the preceding one.

3.6 Final Model Training and Prediction

Finally, equipped with the optimal set of hyperparameters, we proceeded to train our meticulously designed ensemble, the *Boosting on Error* strategy. This ensemble strategy is underpinned by a sequence of models, namely the Random Forest at the base, followed by a series of LightGBM models, and then concluded with two levels of XGBoost regressors, each refining the residuals of its predecessor.

Training proceeded as follows:

1. We trained the Random Forest, our foundational model, on the entire dataset, deliberately excluding rows harboring NULL values.
2. Residuals, or errors, between the true values and the predictions from this base model were computed.
3. For each predictive label, a dedicated LightGBM model was trained on these residuals, subsequently refining them.
4. The XGBoost model at Level 3 was trained next, targeting the residuals from the LightGBM models.
5. The final layer, another XGBoost model at Level 4, was trained to further refine any existing residuals.

To safeguard against overfitting and ensure the model's capability to generalize, we employed rigorous cross-validation techniques throughout this training process [27].

Upon successful training, the ensemble was tasked with generating predictions for the test set. These predictions were then subjected to evaluation using the Mean Column-wise Root Mean Squared Error (MCRMSE), in alignment with the competition's stipulated guidelines.

4 Experiments

The following section describes in detail the steps taken and the modeling of the experimental setup. We begin by describing our dataset and the features we are dealing with. Moving on to the evaluation methodologies, we explain the evaluation metric provided to us for testing and grading and the evaluation metric used for the training. Then we detail the steps taken for training the model along

with data preprocessing, describing how we implemented the mitigating strategies that we planned for the difficulties. The experimental setup will then give us more details on the different setups that we explored for the various sub-parts of the problem. Finally, we conclude the section with the results and analysis of the trained models when used for inferencing and give a comparative study between the capacities of the models.

4.1 Data

The dataset for this competition consists of training and test sets generated from Demographic and Health Surveys (DHS) conducted in 59 countries and various satellite imagery sources. The objective is to predict Mean BMI, Median BMI, Unmet Need Rate, Under 5 Mortality Rate, Skilled Birth Attendant Rate, and Stunted Rate. Missing data is represented as NaN values. This dataset offers a unique opportunity to explore the relationship between satellite imagery and DHS data, enabling the development of accurate predictive models for maternal and child health indicators. Leveraging these data sources can contribute to improved interventions and understanding of this crucial domain.

The data used in this competition is derived from Demographic and Health Surveys (DHS) for 59 countries and various sources of satellite images. Additional datasets like the MOSAIKS satellite image data or other external sources are recommended to improve model performance.

4.1.1 Data Sources and Description

- The health indicators in the DHS data are described in the following document: `https://www.dhsprogram.com/pubs/pdf/DHSG4/Recode6_DHS_22March2013_DHSG4.pdf`
- The document at `https://dhsprogram.com/data/Guide-to-DHS-Statistics/Nutritional_Status.htm` details how these health indicators are calculated.

4.1.2 Data Files

- **gee_features.csv:** An 8GB dataset contains the extracted features from Google Earth Engine (GEE) and keys to match it with other data. It includes country names (DHSCC), cluster numbers (DHSCLUST), and year of survey (DHSYEAR). Note that these column names are not the predictive features. It contains features for both the training and test sets. It can be downloaded from the provided link.
- **training_label.csv:** The label dataset for the training set. The DHSID is used to link the labels to features in gee_features.csv. The objective is to predict the following health indicators: Mean_BMI, Median_BMI, Umet_Need_Rate, Under5_Mortality_Rate, Skilled_Birth_Attendant_Rate, and Stunted_Rate. NaN values represent missing data.
- **sample_submission.csv:** A sample submission file in the correct format. The values of the six health indicators should be replaced with your predictions for submission.
- **train.parquet.gzip / test.parquet.gzip:** Train and test datasets which have been cleaned and stored in parquet format which offers faster loading times as compared to CSV files as well as better compression.
- **low_imp_features.joblib:** The features which should be dropped from the original dataset as detected by our base random forest model.

4.2 Data Analysis

One of the salient aspects that emerged from our preliminary data inspection was the geographical concentration of the low and middle-income countries represented within our training labels. As depicted in Fig. 3, a predominant share of this data emanates from specific global regions: sub-Saharan Africa, Latin America, and South Asia.

Delving deeper, the sub-Saharan African region, with its multifaceted challenges related to maternal and child health, is unsurprisingly a major contributor. The intricacies of health dynamics within this region, exacerbated by limited resources and accessibility constraints, render it a critical area of study. Similarly, the Latin American and South Asian regions, each with its unique socio-economic and cultural landscape, present substantial datasets within our collection. The significance of these

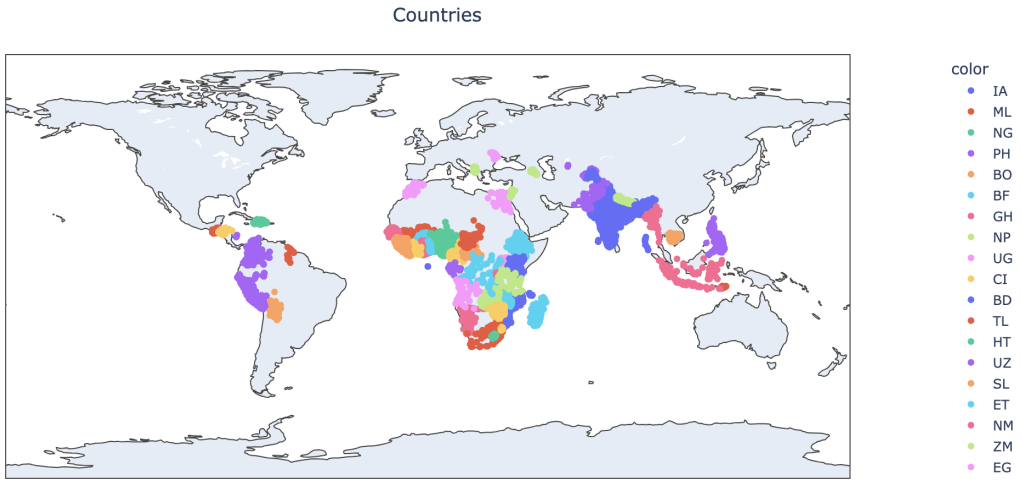

Figure 3: Distribution of the data over the different parts of the World

regions in our study cannot be understated, as understanding the nuances of MCH indicators within these diverse geographies offers insights that could be pivotal for targeted interventions.

It's imperative to note that the bias in geographical representation necessitates careful interpretation of the model outcomes. While the data richness from these regions allows for a robust analysis, the potential for model over-fitting specific to these regions' characteristics exists. As we further our research, it's crucial to ensure that the models are generalizable and adaptable to diverse LMIC contexts beyond the predominant regions in the dataset.

4.3 Evaluation method

The submissions in this competition are evaluated based on the Mean Column-wise Root Mean Squared Error (MCRMSE). The MCRMSE is defined as the average of the individual RMSEs of each predicted column.

4.3.1 Root Mean Squared Error (RMSE)

RMSE is a commonly used measure of the differences between values predicted by a model and the values observed. It is defined as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Where:

- y_i is the original value for each instance i,
- $\hat{y}_i$ is the predicted value,
- n is the total number of instances.

We primarily utilized the Mean Column-Wise Root Mean Square Error to evaluate the models during **training** and **validation** as this was what the leaderboard on our Kaggle competition used to rank submissions.

4.3.2 Submission File

For each DHSID in the test set, the predicted value for each of the six health indicators (Mean_BMI, Median_BMI, Umet_Need_Rate, Under5_Mortality_Rate, Skilled_Birth_Attendant_Rate, and

Stunted_Rate) is required to be provided. The columns should be ordered as mentioned and include a header. The file format is as follows:

```
DHSID,Mean_BMI,Median_BMI,Umet_Need_Rate,Under5_Mortality_Rate,
                    Skilled_Birth_Attendant_Rate,Stunted_Rate
AL200800000997,22.12,22.28,45,6.68,50,20
AL200800000998,20.04,18.98,5.69,7.22,100,10
AL200800000999,18.91,19.47,8.11,0,50,0
etc.
```

This above example describes the file format in which we were required to submit the prediction results on the test set so that the evaluation can be done successfully on the Kaggle platform.

4.3.3 Training

The training regime we adopted encompassed multiple facets: data pre-processing, model selection, hyperparameter tuning, and model validation, all of which were executed on the high-memory infrastructure availed by Google Cloud and AWS EC2 Instances.

During our preliminary data processing, we confronted a myriad of missing values. Rows with even a single Null value were promptly removed, and KNN imputation was subsequently employed, which bestowed us with a refined dataset of over 90,000 data points apt for intensive modeling. The intricacy of managing missing data became evident, more so, given the sheer volume and multifaceted nature of our dataset.

An assortment of machine learning models was appraised, including stalwarts like XGBoost, TabNet, Deep Neural Networks (DNN) as implemented in Pytorch, LightGBM, and Random Forests. Each of these models was put through its paces on a subset of data. The MCRMSE, essentially the RMSE computed across individual columns, provided an incisive lens into the adeptness of each model in predicting specific indicators. This granular assessment was instrumental in iterative model parameter recalibrations.

Hyperparameter tuning was a two-pronged affair. An initial sweep using random search offered a broad-brush perspective, narrowing down potential hyperparameter regions of interest. Subsequently, a detailed grid search was undertaken within these promising areas to meticulously fine-tune the models. This structured approach ensured both breadth and depth in our search for the optimal hyperparameters.

Model/Component	**Hyperparameters**
RandomForestRegressor	n_estimators=2000 verbose=1 n_jobs=-1 random_state=42
LGBMRegressor (per pred_col)	n_estimators=1000 n_jobs=-1 verbose=1 random_state=42
XGBRegressor (level 3)	n_estimators=1000 verbosity=1 n_jobs=-1 random_state=42
XGBRegressor (level 4)	n_estimators=2000 verbosity=1 n_jobs=-1 random_state=42

Table 1: Hyperparameters for the BoostOnErrorEnsemble Model

Turning to the realm of ensemble models, we realized that simple averaging of predictions might not suffice. We thus explored stacked and voting ensembles, with each model's output serving as input for another, potentially stronger, model. Grid searches were also executed for these ensemble techniques, focusing on ascertaining the ideal weights and combination mechanisms.

The Boosting on Errors strategy emerged as an innovative approach in our toolkit. The concept involved sequentially leveraging a series of models: beginning with a Random Forest base, progressing to various LightGBM models, and culminating with multiple XGBoost regressors. Each model in this chain aimed to correct the residuals or errors of its predecessor, thus iteratively enhancing prediction accuracy. Hyperparameter tuning was particularly intricate for this ensemble due to its multi-stage nature.

For model validation, we invoked the tried-and-tested cross-validation technique. Our dataset was sectioned into k fragments. The ensuing model training occurred k times, with each iteration earmarking a distinct fragment as the validation set while utilizing the remainder for training. The k RMSE scores' mean served as the definitive validation metric.

Utilizing the crème de la crème of hyperparameters, as gleaned from the grid search, the final models were trained on the complete dataset (excluding NULL rows). These models then embarked on their prediction journey for the test set. RMSE acted as our performance barometer, guiding any requisite model adjustments.

In retrospection, this training odyssey underscored the quintessence of meticulous data processing, judicious model selection, strategic hyperparameter tuning, and the adoption of innovative ensembles and methodologies like Boosting on Errors, all converging towards achieving predictive excellence in the task at hand.

4.4 Experimental Setup

Our experimental setup was primarily driven by the need to handle a large dataset effectively. We used **dask**, a distributed database package, to process and analyze the data out-of-memory using lazy evaluation. This setup ensured that the system's memory wasn't a bottleneck during our analysis.

4.4.1 Data Preprocessing

Data preprocessing stands as the cornerstone of any machine learning experiment. It was imperative for us to ensure our dataset was not just voluminous, but also of high quality. Capitalizing on the computational efficiency of the **dask_ml** package, we embarked on a dimensionality reduction quest using Principal Component Analysis (PCA). Specifically, the $incrementalPCA$ technique shed light on a salient observation: a mere 728 components encapsulated an impressive 95% of the data variance. This enabled us to significantly cut down on redundant dimensions without compromising the informative essence of the dataset.

Parallelly, we recognized the need to achieve a consistent scale across features, thereby employing normalization techniques. Columns riddled with null values, totaling 500, were promptly eliminated, ensuring the integrity of our dataset. Additionally, adopting a variance threshold strategy, we identified and jettisoned 100 columns which exhibited a variance of less than 1%. This prudent step played a dual role - not only did it eliminate potential noise, but it also streamlined the dataset for swifter computational operations.

Post these rigorous cleansing and reduction operations, our dataset crystallized into a matrix with 728 features spanning 99111 rows. Each of these rows was uniquely identified by DHSIDs. Notwithstanding these efforts, we were confronted with a persistent challenge - missing values in our dependent variables or labels. Rather than discarding these valuable data points, we opted for a more innovative approach. Employing the KNN Imputation method, particularly with a parameter of $n\_neighbors = 75$, we revitalized the dataset by estimating missing labels. This technique, although counter-intuitive at first glance, bore significant fruits. It ensured that vast swathes of potentially insightful data, which would conventionally be sidelined, were retained and actively utilized during model training. This strategy underlines the notion that sometimes, unconventional methodologies can pave the way for enriched datasets and, consequently, superior modeling outcomes.

4.4.2 Model Training and Configuration

The modeling journey began with experimenting on a slew of algorithms. Our experimentation space spanned over a variety of architectures and paradigms, but the most promising results emanated from the Random Forest model. For preprocessing, we utilized the concatenated dataset from `gee_features.csv` and `training_label.csv`. Recognizing the heterogeneous nature of our dataset, we applied one-hot encoding to the categorical variables (DHSCC, DHSREGNA, URBAN_RURA), resulting in a massive, yet comprehensive feature space comprising 13016 distinct features.

The Random Forest model, in its native configuration, furnished a commendable MCRMSE score of 11.32534. While this score was certainly competitive, our objective was to push the boundaries further. Thus, we embarked on a meticulous process of hyperparameter tuning, specifically focusing on pivotal parameters such as $n\_estimators$, $max\_features$, $bootstrap$, and max_depth. This exercise bore fruit, with the model registering an enhanced MCRMSE score of 11.23071, achieved with the hyperparameters set to $n\_estimators = 1200$ and $max\_features = 0.6$.

However, the realm of ensemble modeling beckoned us to explore further. Leveraging the potency of ensemble methods, we ventured into Stacking and Voting Ensembles. The strength of ensemble methods lies in amalgamating the predictive prowess of diverse models, yielding a collective intelligence that often surpasses individual model performances. This was evident when our Stacking and Voting Ensemble configuration pushed the MCRMSE score down to a remarkable 10.94.

Our relentless quest for optimization then steered us towards the avant-garde *Boosting on Errors* methodology, a paradigm that emerged from our collaboration with MLJAR's AutoML platform. This approach revolves around a sequential error correction strategy, where each model in the hierarchy learns from the errors of its predecessor. This ensemble, optimized for error rectification, achieved an outstanding MCRMSE score of 10.759. This evolution from the base Random Forest model to an intricate error-boosted ensemble underscores the significance of iterative modeling and the confluence of diverse techniques in pursuit of model optimization.

4.4.3 Feature Selection and Final Model Training

The initial breakthrough in our modeling approach arose from a meticulous analysis of feature importance derived from the Random Forest model. By capitalizing on the `feature_importances_` attribute of this model, we instituted a threshold criterion, retaining only those features with an importance measure of $>= 0.1\%$. This prudent selection streamlined our feature set to 70 cardinal features, each bearing substantial significance in determining the outcome.

Equipped with this distilled set of features, we trained a Random Forest model adopting a comprehensive set of hyperparameters, notably $n\_estimators = 8000, max\_features = 0.5$, and $max\_depth = 20$. This model registered an impressive MCRMSE score of 11.11011, serving as a testament to the potency of meticulous feature selection and adept hyperparameter tuning in model optimization.

However, our pursuit of excellence did not halt there. Recognizing the power of ensemble techniques, we introduced the Stacking and Voting Ensemble methodologies. While individual models, including our initial Random Forest, exhibited commendable performance, the ensemble approaches, by capitalizing on the strengths of multiple models, ushered in a new pinnacle of performance. Our Stacking and Voting Ensemble model pushed the envelope further by recording an MCRMSE score of 10.94, highlighting the utility of model diversification in performance enhancement.

Yet, our relentless exploration led us to an even more sophisticated modeling strategy, termed *Boosting on Errors*. This methodology emanated from our collaboration with the AutoML platform provided by MLJAR. Instead of solely relying on the predictions of an ensemble or individual models, this approach focuses on sequentially correcting the prediction errors made by the models in the ensemble. It's a technique where each subsequent model endeavors to correct the mistakes of its predecessor. Deploying this innovative strategy, we achieved a superior MCRMSE score of 10.759, marking a watershed moment in our modeling journey. This final score underscores the merit of not just relying on traditionally powerful models, but also exploring novel strategies that challenge conventional wisdom and leverage errors as learning opportunities

4.5 Results

The best results obtained with each model that has been experimented with have been tabulated in the table below. The first column shows the model, and the second column provides the final MCRMSE score of the model on the test set after submission on the Kaggle platform,

Model	MCRMSE
LightGBM	21.89219
DNN (Pytorch)	19.76522
XGBoost	15.46517
CNN (Satellite Images)	12.27347
TabNet	11.8768
Random Forest	11.11011
Weighted Ensemble	10.94367
Boosting on Error	**10.75918**

5 Analysis

Below, we present our observations and analyses derived from the results and training processes, focusing primarily on the improvements and setbacks observed through the column-wise RMSE score evaluation:

- The Random Forest model showcased exemplary performance, producing the lowest Mean Column-wise Root Mean Squared Error (MCRMSE) score of 11.11011. This score, superior to its counterparts, reaffirms the efficacy of ensemble methods, particularly when managing large and diverse datasets.
- XGBoost, known for its gradient boosting capabilities, managed an MCRMSE score of 15.46517. While XGBoost has been hailed for its stellar performance across a plethora of machine learning tasks, it couldn't surpass the Random Forest's metrics for this particular challenge. This accentuates the necessity of tailoring model selection to the specificities of each dataset and task.
- The Deep Neural Network (DNN) registered an MCRMSE score of 19.76522, positioning itself mid-range among the models tested. Such a result hints that even though deep learning methods are capable of discerning intricate patterns, they might not always be the primary choice for every machine learning endeavor. The domain of deep learning presents potential for enhancement, especially when predicting null values based on non-null data, followed by the application of other machine learning models for final regression – an avenue we are keenly pursuing.
- LightGBM, despite its prowess in managing large datasets and computational efficiency, yielded the least impressive score of 21.89219. This drives home the point that model efficiency is invariably intertwined with data characteristics and the nature of the problem.
- TabNet, celebrated for its interpretability, logged an MCRMSE of 11.8768. Even though its performance isn't top-tier, its interpretative capabilities can be invaluable for model comprehension and subsequent enhancements.
- Both XGBoost and LightGBM seemed to struggle in deciphering the intricate relationship between independent and dependent variables in our high-dimensional dataset. This is where random forests demonstrated their superiority. A significant portion of the data columns contained redundant information, rendering traditional dimensionality reduction techniques ineffective.
- A notable characteristic of Random Forest emerged when it began to overfit for n_estimators $>= 8000$. Achieving the optimal combination of parameters was possible only via K-Fold Cross Validation combined with Grid Search. This meticulous approach ensured a model robust enough to avoid overfitting while delivering superior test results with a MSRMSE score of 11.011.
- Our subsequent breakthroughs hinged on ensemble strategies. The Stacking and Voting ensembles further refined our model's precision, recording an MCRMSE score of 10.94.

This technique capitalizes on the collective strengths of multiple models, enhancing the final predictive power. This also empowers that ensembling several models by weighing them according to their learning capacity improves the generalizability of the model and hence the performance.

- Diving deeper into ensemble approaches, our experimentation with the 'Boosting on Errors' methodology facilitated by MLJAR's AutoML yielded the most commendable score of 10.759 MCRMSE. This represents a 3.2% improvement over our previous best. Boosting on Errors, as the name suggests, focuses on instances where our model previously faltered, progressively refining its predictive accuracy.

These findings underscore the critical importance of iterative model refinement, especially when leveraging advanced ensemble strategies and tuning techniques.

5.1 Analysis of Model Performance on Predictive Labels

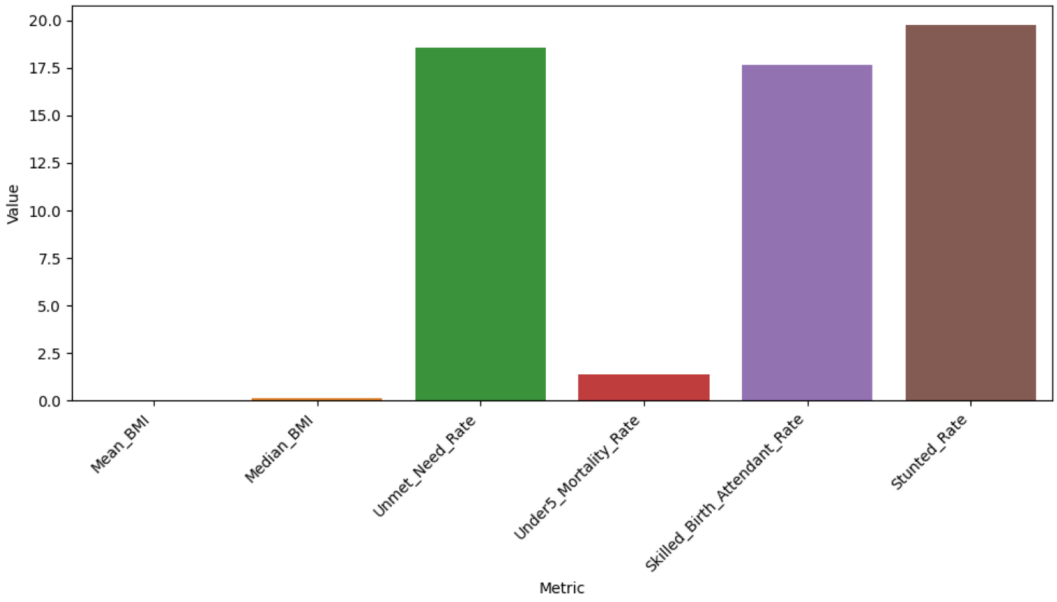

Figure 4: Performance of the best ensemble model over different predictive labels

Upon detailed scrutiny of the model's performance across different predictive labels (Fig. 4), distinctive patterns and insights emerge. The results from the validation set, in terms of RMSE scores for each label, are as follows:

- **Mean_BMI**: Our model showcased exemplary performance with an RMSE of 0.023. This low score implies that our model was highly accurate and successful in predicting the Mean BMI, which reflects the general nutrition level of the population.
- **Median_BMI**: The prediction for Median BMI also portrayed substantial precision with an RMSE of 0.109. Being slightly higher than the Mean BMI, it suggests that there were a few more discrepancies, but the model still captured the central tendency of the BMI values quite well.
- **Unmet_Need_Rate**: With an RMSE of 18.57, this label witnessed higher error. It indicates that predicting the unmet need rate, which can be influenced by a multitude of socioeconomic and cultural factors, posed a more intricate challenge for the model.
- **Under5_Mortality_Rate**: The model delivered a commendable RMSE score of 1.38 for this metric. Given that the Under-5 Mortality Rate is a key indicator of child health and overall development in a region, the model's aptitude in this prediction is particularly significant.
- **Skilled_Birth_Attendant_Rate**: A score of 17.65 RMSE for this metric suggests that while the model was reasonably accurate, there remains room for improvement. This rate is an

indicator of maternal health services and can vary significantly based on the availability and quality of healthcare infrastructure.

- **Stunted_Rate**: The RMSE for predicting stunting rate was 19.78, indicating it as one of the more challenging labels for our model. Stunting, as an indicator of chronic malnutrition, can be influenced by an array of factors including diet, health services, and socioeconomic conditions.

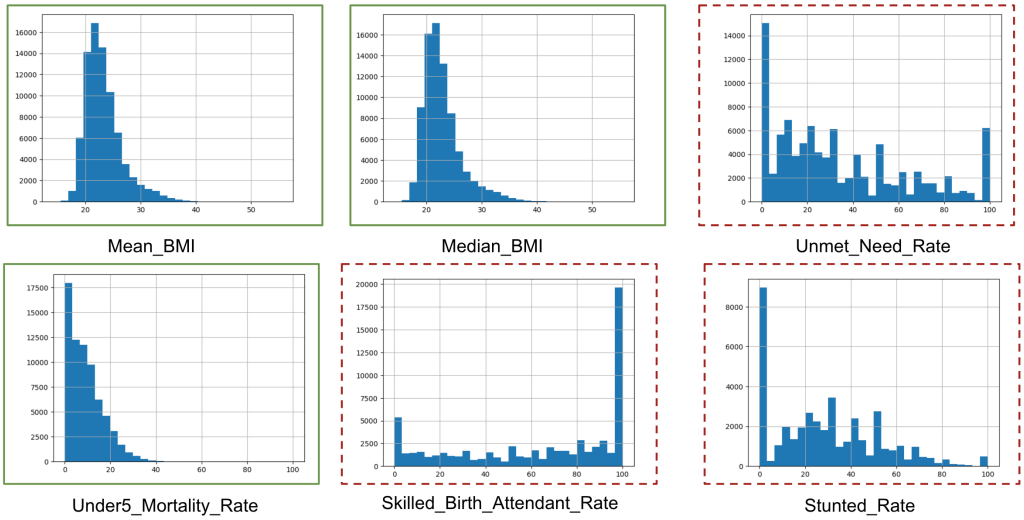

Figure 5: Distribution of the Training Data over different predictive labels

It's crucial to note that the performance discrepancies across labels can be attributed to the volume and nature of available data. Some labels, especially the ones with higher error rates, suffer from data paucity. Moreover, a conspicuous skewness towards the values of 0 and 100 was observed in these labels (Fig. 5), with a dearth of data in the intermediate distribution. This skewness could contribute to the elevated error rates as the model struggled to generalize for the sparse intermediate values. Addressing these data challenges could pave the way for further enhancing the model's predictive prowess across all labels.

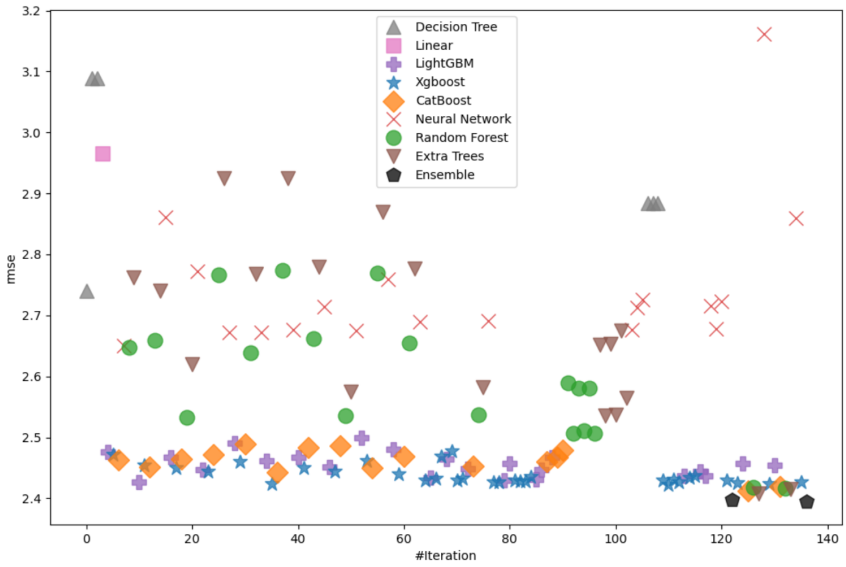

Figure 6: Comparision of the performance of our final ensemble model with the other state of the art models

In our analysis, one of the standout revelations is vividly captured in Fig. 6. This figure delineates the comparative performance of various machine learning models, highlighting the supremacy of our ensemble-based approach—specifically, the "Boosting on Error" strategy. When juxtaposed against standard machine learning models such as LightGBM, Random Forest, XGBoost, CatBoost, and Neural Network, the ensemble model conspicuously stands out in terms of performance. With an RMSE of 2.34, the ensemble model doesn't just marginally outpace its counterparts; it establishes a pronounced lead. Even when comparing it to the next best-performing model, the superiority of the ensemble approach is evident. This stark differentiation underscores the potential of ensemble techniques, especially in complex, high-dimensional tasks where singular models might falter in capturing the intricate nuances of the data.

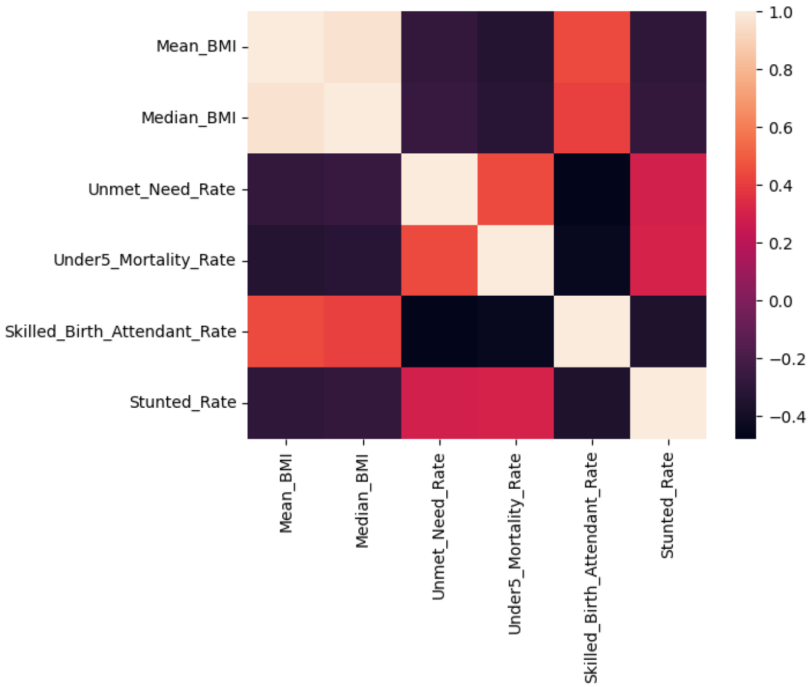

Figure 7: Correlation Matrix over different predictive labels

5.2 Harnessing the Correlation between Mean and Median BMI

In our data exploration phase, a pertinent observation was the high correlation between the Mean BMI and Median BMI values (Fig. 7) to be predicted. This presented an intriguing hypothesis: if the two values are highly correlated, we might be able to leverage the relationship between them to refine our model's predictions.

To test this hypothesis, we crafted a unique approach. Instead of training the model directly on the Median BMI values, we trained it on the difference between the Mean and Median BMI values. Post-training, the model's prediction was then utilized to compute the Median BMI as:

$$Median\_BMI = Mean\_BMI + Model\_Prediction$$

Upon employing this methodology, there was a slight, yet notable improvement in our performance metric. The score improved from 10.761 to 10.759. Although the marginal improvement might seem trivial at first glance, it carries deeper implications. It reaffirms our observation regarding the correlation between the Mean and Median BMI. Furthermore, it signifies the potential of harnessing intrinsic correlations within the data to refine and potentially enhance model predictions. Future work could further delve into similar intrinsic correlations to discover more optimization opportunities.

6 Conclusion

This research firmly accentuates the versatility and power of machine learning in addressing intricate and large-scale health prediction challenges, specifically in low- and middle-income countries. Counter to prevailing preferences for gradient-boosted trees, our evidence showcases that an astutely-tuned random forest model is not only competitive but can indeed outstrip such sophisticated methodologies. This observation echoes the principles of Occam's Razor, which favors simplicity in modeling practices.

A salient takeaway from our work is the paramount importance of **feature engineering**. Through meticulous feature selection, our endeavor realized a remarkable enhancement in the Random Forest's performance, dropping the MCRMSE from 11.32534 to 11.11011. This was made feasible solely by emphasizing the most impactful 100 features, as elucidated by our model's intrinsic characteristics.

Our research also bears testament to the power of ensemble techniques. We noted an even further reduction in MCRMSE to 10.759 by `Boosting on Errors`, underlining the potency of ensemble models, especially when hinged on error correction and stacking methodologies. Moreover, our innovative strategy to exploit the high correlation between the Mean and Median BMI, further honed our model's predictive capability, however marginally.

Beyond the granular technical achievements, this study carries weight in a more expansive context. Our team proudly occupies the **first position** on the public leaderboard, a testament to our novel blend of traditional algorithms and pioneering feature engineering techniques. To date, the uniqueness of our approach remains unparalleled, with no other team matching our performance.

Looking ahead, while our current methodologies have proven robust and effective, there's an ever-present opportunity for refinement. Future work could delve deeper into harnessing intrinsic correlations within datasets, experimenting with newer machine learning models, and potentially integrating domain-specific knowledge to enhance model interpretability.

6.1 Future Work

The following described are some of the innovative areas where we plan to continue our exploration. A lot of these ideas have been influenced by prior works in this field along with the ideations for the mitigating factors for the difficulties faced during the current modelling.

6.1.1 Implementation of a novel RNN/attention-based deep learning model

While our current models have produced competitive results, there is still room for improvement. One possible avenue is to develop a Recurrent Neural Network (RNN) or attention-based deep learning model specifically tailored to our data. Such models are particularly effective at capturing temporal dependencies and may offer superior performance for time-series data that our current models could overlook. There has been significant improvement in transformer-based series prediction networks as well, which is also worth exploring on the dataset.

Figure 8: Examples of NASA Landsat satellite images for (a) Densely (b) Sparsely populated regions

6.1.2 Scraping and utilization of NASA Landsat data

A compelling avenue for enhancing our models is the assimilation of auxiliary data sources. Notably, we advocate for the incorporation of NASA Landsat satellite imagery. Such high-resolution data, teeming with intricate spatial features, holds immense potential to enrich our dataset. By deploying Convolutional Neural Networks (CNNs) on these images, we can leverage the network's capability to discern and learn spatial nuances. Our preliminary endeavors in this direction involved the implementation of a basic CNN architecture, which yielded a promising MCRMSE score of 12.273. While this marks an encouraging starting point, there's ample scope for refinement. Introducing sophisticated image feature extractors, such as the ResNet-50 architecture, could further amplify the efficacy of our model. An illustrative glimpse into the NASA Landsat imagery we intend to harness is provided in Figure 8.

7 Team contributions

Debajyoti Dasgupta: Started with exploratory data analysis and analyzing some models to figure out the best-performing models. Primarily worked with Random Forest Regressor and Deep Neural Networks. Extended on the Random and Grid Search for Random forest regressor hyperparameter tuning. Additionally explored NASA datasets and web scraping to collect image-based data. Was also involved in creating a preliminary CNN-based model and ideation for present and future approaches. Majorly designed, organized, and completed the majority of the report. Additionally performed Grid Search over various ensembling method including Voting and Stacking regression to figure out which model is able to generalize better. Analyzed the data imbalances and worked with AutoML based methodologies to propose and Fine Tune the Boosting on Error Methodology. Designed the final model class for the team and significantly contributed to the Final Report.

Rushil Venkateswar: Preformed dataset cleaning and exploration of Dask methodology and other distributed data packages along with the ideation and model creation and testing for XGBoost, TabNet, and Random Forest model. Significantly contributed to the feature reduction step and efficient data storage with parquet-based storage methods. Also worked on performing Grid search and performing hand tuning hyperparameter tuning to improve the performance of the model. Involved in the creation of the final kaggle notebook and implementation of our model pipeline.

Piran Karkaria: Helped with the exploration and detailed description of the dataset and exploring other sources. Provided insights on data cleaning and writing a first draft of the report.

References

[1] Carla AbouZahr, Don de Savigny, Lene Mikkelsen, Philip W Setel, Rafael Lozano, and Alan D Lopez. Civil registration and vital statistics: progress in the data revolution for counting and accountability. *The Lancet*, 386(10001):1373–1385, 2015.

[2] David E Phillips, Rafael Lozano, Mohsen Naghavi, Charles Atkinson, Diego Gonzalez-Medina, Lene Mikkelsen, Christopher JL Murray, and Alan D Lopez. Are well functioning civil registration and vital statistics systems associated with better health outcomes? *The Lancet*, 386(10001):1386–1394, 2015.

[3] Ties Boerma and Carla AbouZahr. Counting births and deaths 3: Civil registration: why counting births and deaths is important. *The Lancet*, 386:1373–1385, 2014.

[4] Jon Pedersen, Simon I Hay, and Andrew J Tatem. Combining national survey with facility-based data to increase accuracy and spatial resolution of neonatal mortality estimates in uganda. *Health & Place*, 57:187–195, 2019.

[5] Neal Jean, Marshall Burke, Michael Xie, W Matthew Davis, David B Lobell, and Stefano Ermon. Combining satellite imagery and machine learning to predict poverty. *Science*, 353(6301):790–794, 2016.

[6] Soma Bhattacharjee, Nadine Schuurman, and Alan Ker. The use of satellite remote sensing data in health geography: A review. *Journal of Geographic Information System*, 13(1):59–71, 2021.

[7] ICF. The dhs program. Online, Demographic and Health Surveys, 2023.

[8] David H. Wolpert. Ensemble learning. *In The Handbook of Brain Theory and Neural Networks*, pages 110–125, 2001.

[9] Mohammed Zakariah. Classification of large datasets using random forest algorithm in various applications: Survey. In *International Journal of Engineering and Innovative Technology (IJEIT)*, 2014.

[10] Cynthia Yeh, Chenlin Meng, Siyu Wang, Anne Driscoll, Edgar Rozi, Pingjun Liu, Jay Lee, Marshall Burke, David B. Lobell, and Stefano Ermon. Sustainbench: Benchmarks for monitoring the sustainable development goals with machine learning. *arXiv preprint arXiv:2111.04724*, 2021.

[11] Jeremy Irvin, Dillon Laird, and Pranav Rajpurkar. Using satellite imagery to predict health. *Stanford University, Department of Computer Science*, 2023.

[12] Sharmila Reddy Nangi and Nantanick Tantivasadakarn. Global health monitoring from satellite data through multi-task learning. *Stanford University, Department of Computer Science*, 2023.

[13] Rishab Gargeya. Health indicators report. *Stanford University, Department of Computer Science*, 2023.

[14] Sauren Khosla, Benjamin Wittenbrink, and Caroline Zanze. Predicting maternal and infant health outcomes in western africa using satellite images. *Stanford University, Department of Computer Science*, 2023.

[15] Donald B Rubin. *Multiple imputation for nonresponse in surveys*, volume 81. John Wiley & Sons, 2004.

[16] Dorian Pyle. *Data Preparation for Data Mining*. Morgan Kaufmann, 1999.

[17] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[18] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems*, pages 3146–3154, 2017.

[19] Sercan O. Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. *arXiv preprint arXiv:1908.07442*, 2020.

[20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[21] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[22] David H. Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[23] Ludmila I. Kuncheva. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2004.

[24] Harris Drucker. Improving regressors using boosting techniques. In *ICML*, volume 97, pages 107–115, 1997.

[25] Isabelle Guyon and Andre Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[26] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. In *Journal of Machine Learning Research*, volume 13, pages 281–305, 2012.

[27] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, volume 14, pages 1137–1145, 1995.