# 4. Materials and Methodology

In brief, this thesis used a variety of socio-economic and health-related feature variables to predict the national MMR estimate for each country, year sample. The method was described in detail in the following chapter and summarised in Figure 2, below. The development of the final, highest performing model involved X major steps.

1. Applying cross-fold validation, feature selection, and missing data removal to generate different train datasets.
2. Train Random Forest, XGBoost, and LightGBM regressors on different versions of the train dataset to be able to predict MMR estimates as part of a missing data and predictive analysis.
3. Train stacking and voting ensemble models on each of these base estimators to improve MMR estimates.
4. Evaluate best-performing model by analysing feature importance and sensitivity to input data as well as comparing its estimates to MMR predictions in the literature.

All code was written using Python3 and run in Visual Code Studio or on the Gadi supercomputer at the National Computational Infrastructure. Where appropriate, the random seed was set to 42 for reproducibility. All datasets and code used in this thesis were uploaded to a public GitHub repository (<https://github.com/R0sle/health_economics_honours>). Unfortunately, some of the datasets were too large to be uploaded. All datasets are available upon request.

A diagram of data analysis

AI-generated content may be incorrect. A diagram of a graph

AI-generated content may be incorrect.

A diagram of a model

AI-generated content may be incorrect.

**Figure 2**: Flowchart of experimental workflow. It gives a high-level overview of the process of generating train, validation and test datasets and fitting a variety of decision-tree based ML models to the input data.

## 4.1 Data Sources and Merging

### 4.11 Data Sources

Data was sourced from a variety of World Health Organisation (WHO) and World Bank Group (WB) data repositories. The final merged dataset consisted of a mixture of socioeconomic, health-related, and environmental indicators from a range of sources. Using data from a variety of data sources was motivated by comments made by Onambele et al. (2023), who recommended that future research into maternal mortality estimates combine data from multiple sources to take advantage of the different datasets offered by the WHO.

The national estimates for the maternal mortality ratio (MMR) were sourced from the World Bank Group’s Gender Data Portal (30). This dataset contains MMR estimates for 242 regions, countries, territories, and areas between 1985 and 2018. **These national MMR estimates served as the ‘ground truth’ in this study.**

Some of the datasets used in this study contained disaggregated data. For example, features were sex or economic status specific. However, the national MMR estimates were not disaggregated. Including the disaggregation as its own feature column would therefore produce a missing value in the MMR estimate column when merging the datasets. To prevent this, I created subgroup specific versions of the feature for each included demographic. See Table 2 for an illustrative example. If the data was disaggregated on a scale (e.g. Feature A was reported for wealth quintiles 1 through 5), I only used values from the most extreme subsgroups (e.g. quintiles 1 and 5) to prevent the number of features, and thus the dimensionality of the dataset, from becoming too large.

**Table 1**: Illustrative example of subgroup specific versions of a single feature, ‘Feature 1’, with the bolded text defining the demographic being represented.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Country | Date | Feature 1 **Female** | Feature 1 **Male** | Feature 1 **Rural** | Feature 1 **Urban** | Feature 1 **Wealth Quintile 1** | Feature 1 **Wealth Quintile 5** |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

Information about the specific datasets used in this study was summarised in Table 2, below. See the Appendix, section 9.1, for the specific variables gathered from each data source. Each of the individual features were collated by WHO technical programs [rep gho meta-data].

**Table 2:** Summary information about the datasets used in this study.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Type of Dataset** | **Number of Features** | **Date Range** | **Number of Areas Covered** | **Demographic Subsets Used** | **Source** |
| National MMR estimates | 1 | 1985-2018 | 242 | NA | World Bank Group’s Gender Data Portal [30] |
| Health outcomes & literacy, agency | 198 | 1960-2023 | 265 | NA | World Bank Group’s Gender Data Portal [31] |
| Illness incidence and prevalence | 193 | 2000-2019 | 194 | Sex | WHO Health Inequality Data Repository & IHME [34] |
| Empowerment | 9 | 1991-2023 | 120 | Economic status (quintiles 1, 5) | WHO Health Inequality Data Repository after re-analysis by the WHO Center for Health Equity Monitoring [32] |
| Socioeconomic, education, environmental variables | 64 | 1970-2023 | 195 | Sex, economic status (quintiles 1 and 5), residence (urban, rural) | WHO Health Inequality Data Repository, sourced from the World Bank Data Catalogue [33] |
| World Bank’s Categorisation of a Country’s Income level | 1 | 2024 |  | NA | WHO Health Inequality Data Repository [35] |

### 4.12 Merging Data

All datasets used in this report contained columns specifying the country and its associated ISO3 country code, as described in the ISO 3166 international standard [<https://www.iban.com/country-codes>]. However, different datasets sometimes used a different version of the same country’s name (e.g. United States versus United States of America). Therefore, the datasets were merged using the unique ISO3 code and year.

The final merged data set had rows uniquely identified by their country and year, with feature columns extracted from the original dataset. This merged dataset contained 16,948 samples and 732 features.

As described in Table 2, the national MMR estimates were collected between 1985 and 2018. Therefore, all data collected before 1985 and after 2018 was excluded. Features with no data between 1985 and 2018 years were also excluded. This restricted the dataset to 9,018 rows and 726 columns.

### 4.13 Data Cleaning

All year/country combinations that were missing national MMR estimate values were removed to avoid imputing the estimate. This imputation may have caused the machine learning models to be trained on incorrect feature/MMR estimate pairings, introducing inaccuracy. As a result, the number of rows in the dataset decreased from 9,018 to 2,789.

I removed the following feature variables:

* 'Number of maternal deaths’
* ‘Lifetime risk of maternal death (1 in: rate varies by country)’
* ‘Lifetime risk of maternal death (%)’

The ‘number of maternal deaths’ is the numerator of the MMR. Similarly, the two features measuring the ‘lifetime risk of maternal death’, as a rate or percentage, are calculated using the MMR [UN MMEIG report]. The Pearson’s correlation coefficient between the MMR estimate and the ‘lifetime risk of maternal death (%) was 0.93, showing their strong positive correlation. Therefore, these three variables were excluded from the feature dataset to prevent the model from using them to predict the MMR instead of learning the relationship between the MMR and socioeconomic and health-related data.

The final data cleaning step involved removing the ‘country’ and ‘year’ as feature variables, instead using them as unique sample identifiers. This step was performed because ‘country’ and ‘year’ could introduce bias in the data, where the model learns the typically MMR for a country or year instead of learning to use the relationships between the MMR and the socio-economic and health-related variables.

As a result of this pre-processing, **the final, merged dataset consisted of 2,789 samples and 721 columns (one output variable and 720 features).**

## 4.2 Exploratory Data Analysis

An initial exploratory data analysis was conducted to gain a better understanding of the dataset and motivate the choice of pre-processing techniques.

### 4.21 Missing Data Analysis

While all samples with missing MMR data were excluded from analysis (as described in section 4.13), it was instructive to explore and discuss the missing data occurrence in the MMR estimates. Thus, the proportion of missing MMR data per year was calculated for each income level between 1985 and 2018 (see Section 5.11). The proportion of missing data per year in the input features was also calculated with the goal of better understanding the input data distribution. The results of this analysis were shown in Figure 3 below, as they motivated how train/test subsetting was conducted.

A graph of a number of missing data

AI-generated content may be incorrect.

**Figure 3:** Proportion of missing data across all countries and features per year between 1985 and 2018.

Before 2000, the dataset had close to or greater than 90% missing data. In contrast, between 2000 and 2018, the dataset had been 40 and 80% missing data. For 4 years, the proportion of missing data was less than 50%. The specific figures were shown below in Table 3. Generally, there was little difference between the proportion of missing feature data across the different income levels.

**Table 3:** Years with the lowest proportion of missing data across all countries and features, rounded to two decimal places.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Proportion of Non-Missing Data** | | | |
| **Low Income (%)** | **Lower-Middle Income (%)** | **Upper-Middle Income (%)** | **High Income (%)** |
| 2000 | 25 | 27 | 26 | 30 |
| 2005 | 32 | 33 | 32 | 35 |
| 2010 | 22 | 25 | 24 | 28 |
| 2015 | 24 | 25 | 24 | 27 |

This pattern was likely due to a group of indicators being reported with a periodicity of 5 years. It was taken into account when splitting the data into train/test subsets, where at least one year of low missing data was used in the test set (see Section 4.241).

No data imputation was used despite the high level of missing data. This was done to prevent imputation methods from introducing bias into the dataset, especially given the high level of missing data. More specifically, the pattern of missing data in this dataset would typically be categorised as ‘missing not at random’ (MNAR). In other words, the probability of the data being missing relies on both the observed and missing data, or on other, unobserved variables [2, 3]. This dataset would be considered MNAR because the probability of missing data is heavily related to the robustness of the country’s data collection systems, which is an unseen variable. In this case, it is usually not possible to handle the missing data, as such handling would require observation of the unseen data [2]. For example, removing all rows and columns with missing data or imputing the missing values based on the observed datapoints could bias the data towards those countries reporting greater amounts of data. As a result, I only used machine learning models that could work missing data.

### 4.22 Key Statistics

To give a deeper insight into the feature dataset, I present key summary statistics about some of the features that the literature describes as having a meaningful relationship with MMR. The proportion of missing data was calculated after data cleaning, explaining why the MMR estimates have a proportion of missing data of 0%. Both mean and median were included to give an indication of outlier occurrence in the feature data. The following discussed was motivated by the results presented in Table 4.

Generally, health outcomes improved as income level increased, as expected. Standard deviation in the outcome decreased as income level decreased, indicating a more uniform quality of care. While many of the important variables have low rates of missing data, some of the important socio-economic and quality of care features had increasing proportions of missing data as income level increased. For example, the dataset for the lowest income level countries was missing 58% of data regarding ‘women participating in own health care decisions (% of women age 15-49)’ while the dataset for the highest income countries was missing 99.9% (see Table 4). This feature was sourced from a Demographic Health Survey (DHS), which is normally collected in lower income countries, explaining the data disparity. This is a good trend to be aware of when analysing feature importance for features measured more frequently using DHS data.

According to Table 4, the national MMR estimates were subject to large outliers, as the mean values were larger than the median values for all income levels. Additionally, the standard deviation for the MMR estimates was large. The difference between mean and median, as well as the magnitude of the standard deviation, decreased as income level increased. This indicates that outliers will be more common for lower income countries, as they have a wider variety of possible MMR values than higher income countries.

**Table 4:** Mean, median, standard deviation and proportion of missing data of features with a meaningful relationship to MMR. The key summary statistics were presented per income level.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature** | **Income Level** | **Mean** | **Median** | **Standard Deviation** | **Proportion of Missing Data (%)** |
| WHO national MMR estimate | Low | 657 | 617 | 453 | 0 |
| Upper middle | 197 | 55 | 260 | 0 |
| Lower middle | 51 | 38 | 55 | 0 |
| High | 15 | 8 | 21 | 0 |
| Infant mortality rate (per 1,000 live births) | Low | 63 | 65 | 29 | 0 |
| Upper middle | 43 | 39 | 23 | 0 |
| Lower middle | 24 | 19 | 15 | 0 |
| High | 9 | 7 | 7 | 2 |
| Pregnant women receiving prenatal care (%) | Low | 74 | 85 | 23 | 28 |
| Upper middle | 81 | 86 | 18 | 65 |
| Lower middle | 92 | 96 | 10 | 78 |
| High | 93 | 97 | 8 | 95 |
| Women participating in own health care decisions (% of women age 15-49) | Low | 55 | 60 | 22 | 58 |
| Upper middle | 65 | 67 | 22 | 86 |
| Lower middle | 84 | 84 | 8.7 | 97 |
| High | 91 | 91 | NaN | 99.9 |
| Communicable, maternal, neonatal, and nutritional diseases prevalence in females (age standardized) (per 100,000 population) | Low | 79,399 | 84,661 | 14,140 | 77 |
| Upper middle | 73,030 | 73,279 | 9,389 | 84 |
| Lower middle | 62,248 | 63,092 | 10,658 | 86 |
| High | 38,835 | 36,807 | 11,821 | 87 |
| Survival to age 65, female (% of cohort) | Low | 59 | 58 | 13 | 0 |
| Lower-Middle | 71 | 73 | 12 | 0 |
| Upper-Middle | 79 | 80 | 8 | 0 |
| High | 87 | 88 | 5 | 0 |
| Unmet need for contraception | Low | 27 | 28 | 7 | 33 |
| Upper middle | 22 | 23 | 8 | 72 |
| Lower middle | 13 | 12 | 6 | 88 |
| High | 13 | 10 | 10 | 97 |

Removing samples missing an associated MMR estimate caused the number of samples from each income level to decrease (see Table 5). The proportion of samples remaining after this data cleaning step increased as income level increased, highlighting the lack of routine data collection services in lower income countries.

**Table 5:** Number of samples per income level before and after samples missing an MMR estimate were removed. The proportion of datapoints remaining per income level after removal of samples with their MMR estimate missing was given as a percentage.

|  |  |  |  |
| --- | --- | --- | --- |
| **Income Level** | **Number of Samples** | | **Proportion of Samples Remaining (%)** |
| **Before Removing Samples with MMR Missing** | **After Removing Samples with MMR Missing** |
| **Low** | 884 | 78 | 8.8 |
| **Lower-Middle** | 1734 | 310 | 17.88 |
| **Upper-Middle** | 1802 | 996 | 55 |
| **High** | 2176 | 1405 | 65 |

### 4.23 Principal Component Analysis

Principal component analysis (PCA) was employed for dimensionality reduction purposes. Rather than trying to visualise patterns by plotting all 721 feature dimensions, the PCA method was used to project the samples into the 10 principal components that captured the maximum amount of total variance across the dataset.

PCA cannot be performed on a sparse dataset. Thus, the scikit learn implementation of k-Nearest Neighbours imputation was performed before applying PCA to the dataset, where missing values were imputed using the values of the most similar data points [4]. The data was standardised to a zero mean and unit standard deviation before applying the scikit learn PCA method [5]. This prevented variance calculations from being skewed by features with high magnitudes [5]. The variance captured by the top ten principal component was calculated and presented in Section 5.12.

As demonstrated by Figure 5, the first principal component captured 31% of total variance in the dataset. The percentage of variance captured decreased sharply to 9 and 6% for the second and third principal components before levelling out at between 1.7 and 3% for the remaining top ten principal components. Thus, using the first two principal components to represent the dataset would capture approximately 40% of the data’s total variance, providing an adequate representation for the purposes of exploratory data analysis.

A graph with blue squares

AI-generated content may be incorrect.

**Figure 5:** Percentage of total variance explained (%) per principal component for the top ten principal components.

The input data was plotted across their two most important principal component axes, with the samples were coloured according to their income level, MMR estimate, and year to better identify patterns and clusters in the data. See Section 5.12 for the results of the PCA analysis.

### 4.24 Correlation Analysis

Analysis of the pairwise correlations between specific features and the MMR estimates was used to inform feature selection strategies. The following statistics motivated the use of correlation-based feature selection strategies (explained in Section 4.33). While there was a broad range of correlations between features and the MMR, the frequency of coefficients was not uniformly distributed (see Figure 6). More specifically, over 50% of features’ pair-wise correlation coefficient with MMR was between -0.25 and 0.25. In contrast, approximately 2% of features had a pair-wise coefficient less than -0.75 or greater than 0.75. The low frequency of high magnitude pair-wise correlations between features and MMR estimates motivated the use of feature selection methods as a potential method of reducing overfitting to noise.

A blue bar graph with black text

AI-generated content may be incorrect.

**Figure 6**: The Pearson’s pair-wise correlation coefficient between a specific feature and the MMR plotted against the proportion of features in the cleaned input dataset with this correlation coefficient.

## 4.3 Data Pre-Processing for Machine Learning Pipeline

Refer to Figure 2 for a flowchart overview of the data pre-processing process.

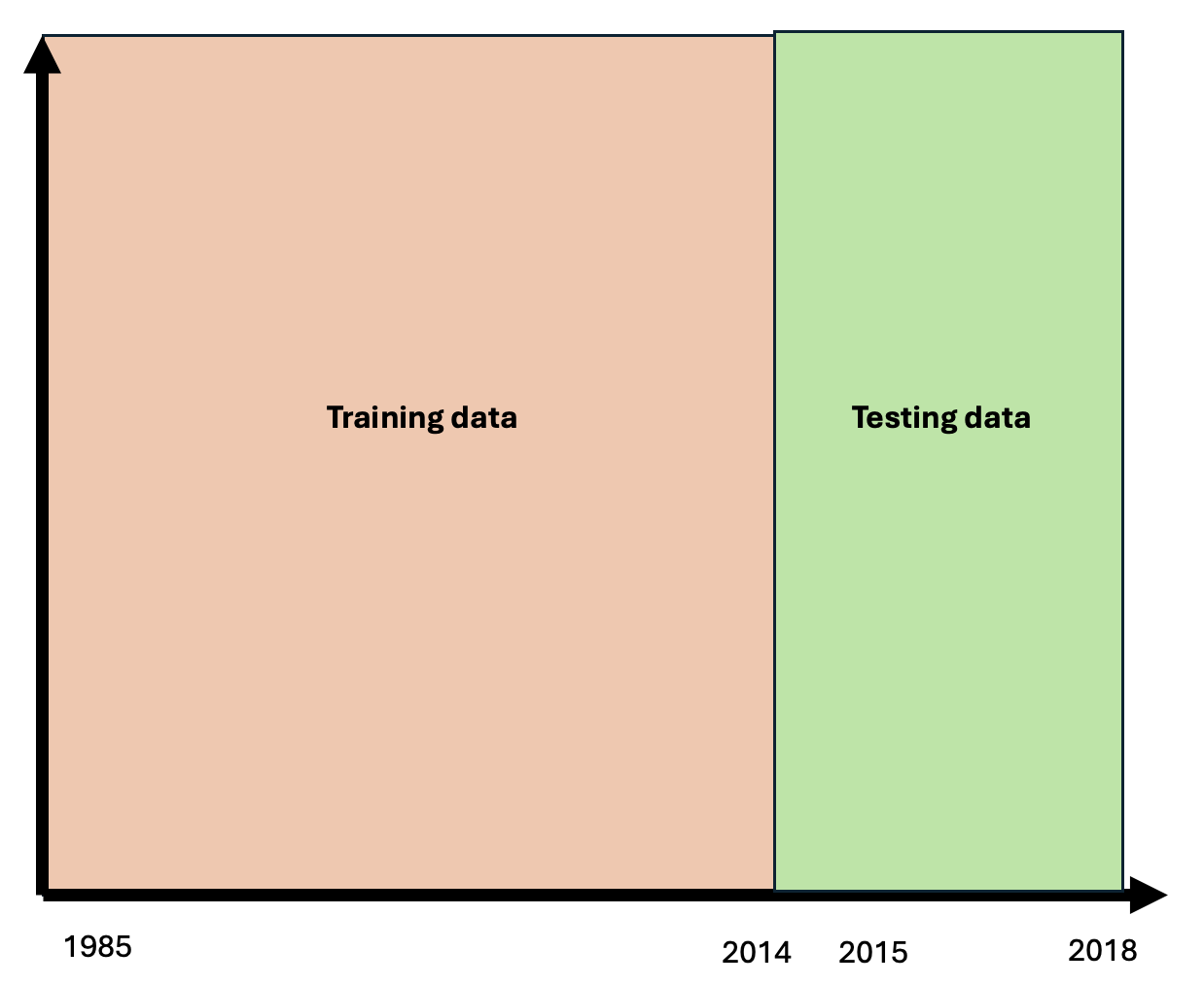
### 4.31 Splitting Input Data into Train Test Sets

The cleaned and pre-processed dataset was split into train and test subsets in two different ways, each for a specific purpose.

A yellow and blue chart

AI-generated content may be incorrect.

a)



b)

**Figure 7:** Visualisation of train/test split for a) missing data analysis and b) predictive analysis.

#### 4.311 Missing Data Analysis (MDA):

The primary aim of this thesis was to predict MMR estimates for countries with sparse data. To achieve this purpose, I split the input data so that the same country could only be in the train or test set. This simulates the scenario where the model is trained on available data from countries with data collection services and is then used to predict MMR estimates for countries with no data. Models trained in this way could inform policy makers about their country’s maternal health status. Additionally, splitting the input data by country prevents data leakage by preserving independence between the train and test sets.

Each country in the input dataset was classified as high-income, upper middle income, lower middle income, or low-income by the World Bank. I split the original input dataset into 4 subsets, each containing all countries from a specific income level. Each of these subsets was split into train/test datasets in a 90:10 ratio. The 4 train sets were then merged into one complete train set, and the 4 test sets were merged into a complete test set. This procedure ensured that rather than risking all samples from a particular income level being in just the train set, thus ensuring that the test set can evaluate the model’s generalisability.

However, ensuring that all data from the same country was only in the train or test set meant the train/test split was not exactly 90:10. Instead the true ratio was within one or two percentage points of 90:10, as the number of rows per country did not allow for an exact 90:10 split.

#### 4.312 Predictive Analysis (PA):

This thesis also aimed to train models to forecast MMR estimates. These forecasts could give policy-makers information about future MMR trends as well as allow researchers to forecast the effects of candidate policies. I simulated this scenario by placing all data from 1985-2014 in the train set and all data from 2015-2018 in the test set. As a result, the model would learn patterns between consecutive years and would be evaluated on its ability to use these patterns to predict the future.

This division produced an 88:12 train/test split. It was not a clean 90:10 split because I included data from 2015 in the test set. I did this to ensure that the test set contained sufficient non-missing data to be useful, as 2015 was one of the four years containing less than 50% missing data (see Section 4.21).

### 4.32 Cross-Validation

Each of the MDA and PA train datasets were further divided into 5 cross-validation folds, each of which being a specific permutation of the train/validation 80:20 split.

If the data was being split for MDA, the same country was placed in either the train or validation set. This prevents data leakage and evaluates the model’s ability to predict sparse country data. If the data was split for PA, the same year was placed in either the train or validation set to prevent data leakage and assess the model’s ability to predict for an unknown year. These conditions resulted in slight deviations from the 80:20 ratio, but only within a percentage point.

The train data was split into cross-validation folds using Scikit Learn’s GroupKFold method, which ensures that entries of the same group only appear in one validation set [15]. Members of the groups were countries for MDA and years for PA. This method ensures that the train and validation sets differ between the different folds, allowing me to assess how changes in the composition of the training dataset affected model performance.

### 4.33 Feature Selection

As discussed in the literature review, decision-tree based models can work with high-dimensional data. However, I tested whether their ability to work with many dimensions meant that they achieved the greatest performance when using high-dimensional data, or if they achieve greater performance when working with a subset of features. This difference may be due to a higher number of features introducing noise, sparsity, and additional computational complexity, as discussed in Section 3.2.

I created 5 versions of each fold, each with a different selection of features (see Table 6 for summary and the GitHub repository for a spreadsheet giving the specific features used in each subset). In the first case, no features were removed to evaluate model performance on the full feature dataset, which contained 720 features.

To create the second feature subset, I surveyed a number of papers about maternal mortality to learn about which features researchers believed most strongly influence MMR [7, 8, 9, 10, 11, 12, 13, 14]. As a result, I hand-picked 40 biological and socio-economic feature variables. While there were many more relevant features I could have chosen from the available dataset, I believed these covered the major maternal mortality determinants.

The final three feature subsets were produced using the correlation between feature variables and MMR. More, specifically, I computed the pairwise Pearson’s correlation coefficient of all feature columns with the MMR estimate. I used the Pandas correlation method, which ignores rows where either feature pair has a missing value. I created a dataset containing features whose absolute pairwise correlation coefficient with MMR was at least 0.8. I produced two additional datasets containing features whose absolute correlation coefficient with MMR was at least 0.7 and 0.6. This allowed me to test the strength of correlation needed for the features to improve model performance.

**Table 6:** Summary of the 5 different feature selection methods used.

|  |  |
| --- | --- |
| **Feature Selection Method**  **Description of Feature Selection Method** | **Number of Features** |
| No feature selection employed. | 720 |
| Performed a literature review of the features believed to have the highest influence on MMR [7, 8, 9, 10, 11, 12, 13, 14]. These features were hand-picked. | 40 |
| Included all features with an absolute Pearson’s pairwise correlation coefficient with MMR >= 0.8 | 11 |
| Included all features with an absolute Pearson’s pairwise correlation coefficient with MMR >= 0.7 | 45 |
| Included all features with an absolute Pearson’s pairwise correlation coefficient with MMR >= 0.6 | 113 |

### 4.34 Iterative Removal of Rows and Columns with a Higher Proportion of Missing Data Than a Specific Threshold

As discussed in Section 4.21, my input data set contains information missing not at random. In this case, imputation and/or data removal can introduce bias, as the presence of missing data may signal important information about the country’s health system dynamics. However, if the specific feature variable has a very high proportion of missing data, such as over 85%, the model may overfit to its small amount of data.

Therefore, I experimented with removing rows and columns with very high proportions of data. More specifically, I iteratively removed columns and rows from the dataset that contained a higher proportion of missing values than a pre-defined threshold. Higher thresholds meant rows and columns with a greater proportion of missing values were kept in the dataset. Given that removal of a sparse row could affect the proportion of missing data in a column, and vice versa, the removal of rows and columns with more missing data than the specified threshold was conducted until the dataset stabilised.

The number of columns influenced the proportion of missing data per row, affecting the iterative thresholding operation. Since each feature subset had a different number of columns, the iterative removal of missing data had to be performed per feature subset/fold pair. It was performed per fold instead of on the entire feature subset-specific train set to prevent data leakage between the folds.

Iterative missing data removal was only applied to the training data, not the validation or testing data. This allowed evaluation results from different training datasets to be compared, as each evaluation dataset was at most a subset of the others. However, to ensure model compatibility, any columns dropped from a model’s train set due to feature selection or missing data removal were also dropped from the validation and test sets.

I produced 4 versions of each feature subset/fold combination using the following thresholds:

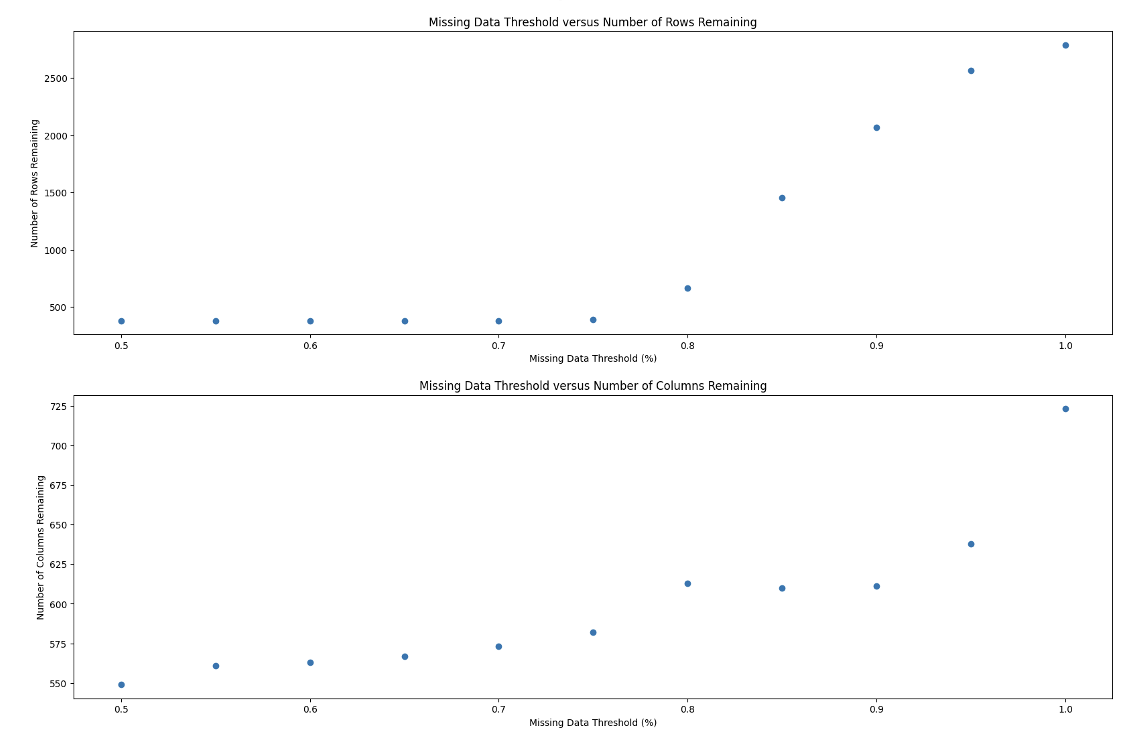
* All columns and rows missing more than 85% of their data
* All columns and rows missing more than 90% of their data
* All columns and rows missing more than 95% of their data
* No missing data removed (100% threshold)

See Figure 8 for a general idea of how iterative data removal affected the input dataset’s size. My lowest missing data threshold of 85% still preserves a fair amount of missing data (61%), staying true to the data sparse conditions of countries without robust data collection systems. Higher missing data thresholds retained a larger number of rows and columns. Decreasing the missing data threshold from 95% to 90% had a large impact on the number of rows (2568 to 2070) but only a small effect on the number of columns (611 to 610).

A graph with blue dots

AI-generated content may be incorrect.

a)



c)

b)

**Figure 8:** The a) proportion of missing data remaining, b) number of rows and c) number of columns remaining in the full input set (not split into train/validation folds or feature subsets) after iterative data removal for missing data thresholds between 50% and 100% (no missing data removal).

Stricter missing data limits were not employed due to the possibility of introducing bias via removing missing data when data is not missing completely at random, as discussed in Section 4.21. Additionally, smaller missing data thresholds would have reduced the size of the dataset to 700 rows, which would produce small datasets prone to overfitting.

### 4.35 Summary of Datasets Produced Via Pre-Processing

The cleaned data was split into train/test subsets in a 90:10 ratio, with different versions of the split being implemented for missing data versus predictive analysis. These subsets were each further split into 5 train/validation cross-validation folds. 5 versions of each fold were created by applying different feature selection mechanisms. Finally, 4 versions of each of each feature subset/fold combination was produced by applying iterative missing data removal with different thresholds of missing data allowed.

This produced 100 datasets for each of the missing data and predictive analyses.

## 4.4 Models Training and Fine-Tuning

### 4.41 Base Model Development

#### 4.411 Training and Fine-Tuning

Scikit Learn’s Random Forest Regressor, XGBoost’s XGBRegressor and LightGBM’s LGBMRegressor were trained to predict the maternal mortality ratio (MMR) for a specific year, country pair [16, 17, 18]. 100 versions of each model type were trained for missing data analysis, with each version corresponding to one of the 100 versions of the train dataset produced during pre-processing. Another 100 versions of each model type were trained for predictive analysis.

The models’ hyperparameters were fine-tuned using the Optuna hyperparameter optimisation framework [19]. Finetuning occurred over 1,000 Optuna trials, where each trial represented a choice of values for the subset of hyperparameters being tuned. See Tables 7, 8, and 9 for the specific hyperparameters fine-tuned for each model type. All other hyperparameters were set to their default values. During each trial, the model being fine-tuned was fit to its associated train data using the chosen hyperparameter values. Its performance was evaluated by calculating the mean squared error (MSE) of its predictions on data from the associated validation fold. The model with the lowest validation MSE across the 1,000 trials was saved.

This method produced 100 fine-tuned XGBoost, LightGBM, and Random Forest models for the missing data analysis and 100 models for the predictive analysis. Thus, 600 models were fine-tuned in total, with the entire fine-tuning process taking approximately 1 to 2 days. As a result of this computational complexity, fine-tuning was conducted using one metric rather than multiple metrics capturing different information. MSE was used because it heavily penalises outliers, which in this context would most likely refer to errors in the high MMR estimates for lower-income countries.

**Table 7:** Hyperparameter Tuning for Scikit-Learn’s Random Forest Regressor [16]

|  |  |  |
| --- | --- | --- |
| **Hyperparameter Name in Scikit-Learn** | **Hyperparameter Function** | **Range of Potential Values** |
| n\_estimators | The number of trees in the random forest. | 10 to 300 |
| max\_depth | The maximum depth of trees in the model. | 3 to 25 |
| min\_samples\_split | The minimum number of samples/rows for which an internal node can be split. | 2 to 10 |
| bootstrap | Whether each tree in the random forest is trained on a random subset of samples. | True or False |
| max\_samples | The proportion of the full dataset used to train each base estimator. This parameter is not used when bootstrap is set to False. | 0.01 to 1.0 |

**Table 8:** Hyperparameter Tuning for XGBoost’s XGBRegressor [17]

|  |  |  |
| --- | --- | --- |
| **Hyperparameter Name in XGBoost** | **Hyperparameter Function** | **Range of Potential Values** |
| n\_estimators | Number of boosting iterations/number of trees in the final model. | 10 to 300 |
| max\_depth | The maximum depth of trees in the model. | 3 to 25 |
| learning\_rate | Controls the extent to which each new tree influences the model’s predictions. | 0 to 1 |
| reg\_alpha | Constant used for L1 regularisation. | 0 to 0.001 |
| reg\_lambda | Constant used for L2 regularisation. | 0 to 0.001 |
| booster | ‘gbtree’ is the XGBoost gradient boosting. ‘dart’ modifies ‘gbtree’ to randomly drop trees to reduce overfitting. | ‘gbtree’ or ‘dart’ |
| subsample | The proportion of data randomly chosen for each boosting iteration during training. | 0.1 to 1 |

**Table 9:** Hyperparameter Tuning for LightGBM’s LGBMRegressor [18]

|  |  |  |
| --- | --- | --- |
| **Hyperparameter Name in LightGBM** | **Hyperparameter Function** | **Range of Potential Values** |
| n\_estimators | Number of boosting iterations/number of trees in the final model. | 10 to 300 |
| max\_depth | The maximum depth of trees in the model. | 3 to 25 |
| learning\_rate | Controls the extent to which each new tree influences the model’s predictions. | 0 to 1 |
| reg\_alpha | Constant used for L1 regularisation. | 0 to 0.001 |
| reg\_lambda | Constant used for L2 regularisation. | 0 to 0.001 |
| boosting | ‘gbtree’ is the XGBoost gradient boosting. ‘dart’ modifies ‘gbdt’ to randomly drop trees to reduce overfitting. | ‘gbdt’ or ‘dart’ |
| bagging\_freq | Every k-th iteration, the model selects a random subset of data for use in training for the next k iterations. | 0 to 10 |
| bagging\_fraction | The proportion of data randomly chosen for training. Used if bagging\_frequency is not set to equal zero. | 0.1 to 1.0 |

#### 4.412 Testing and Comparison

The hyperparameters for each fine-tuned model were saved at the end of the Optuna fine-tuning process. When being evaluated, each model was re-trained with these hyperparameter values on its associated training data.

The 300 fine-tuned models for missing data analysis were evaluated on the same test set, which had no missing data removal applied to it. The only difference between the test sets used for the various models is that models were evaluated on a version of the test set that contained on the features that were present in the model’s train dataset.

The fine-tuned models were evaluated based on the accuracy of their test set predictions. More specifically, the MSE, root mean-squared error (RMSE), mean absolute error (MAE), R2, and mean relative error of its test set predictions were calculated. Using a wide range of metrics enabled a more nuanced evaluation of the model’s performance, as the metrics placed different emphasis on outliers. The mean relative error was a symmetrical version of the mean absolute percentage error (MAPE) described in Section 2.3211. The formula for the mean relative error is given in Equation 11, below, with the modification to the base MAPE score (Equation 5) adjusting for its asymmetrically.

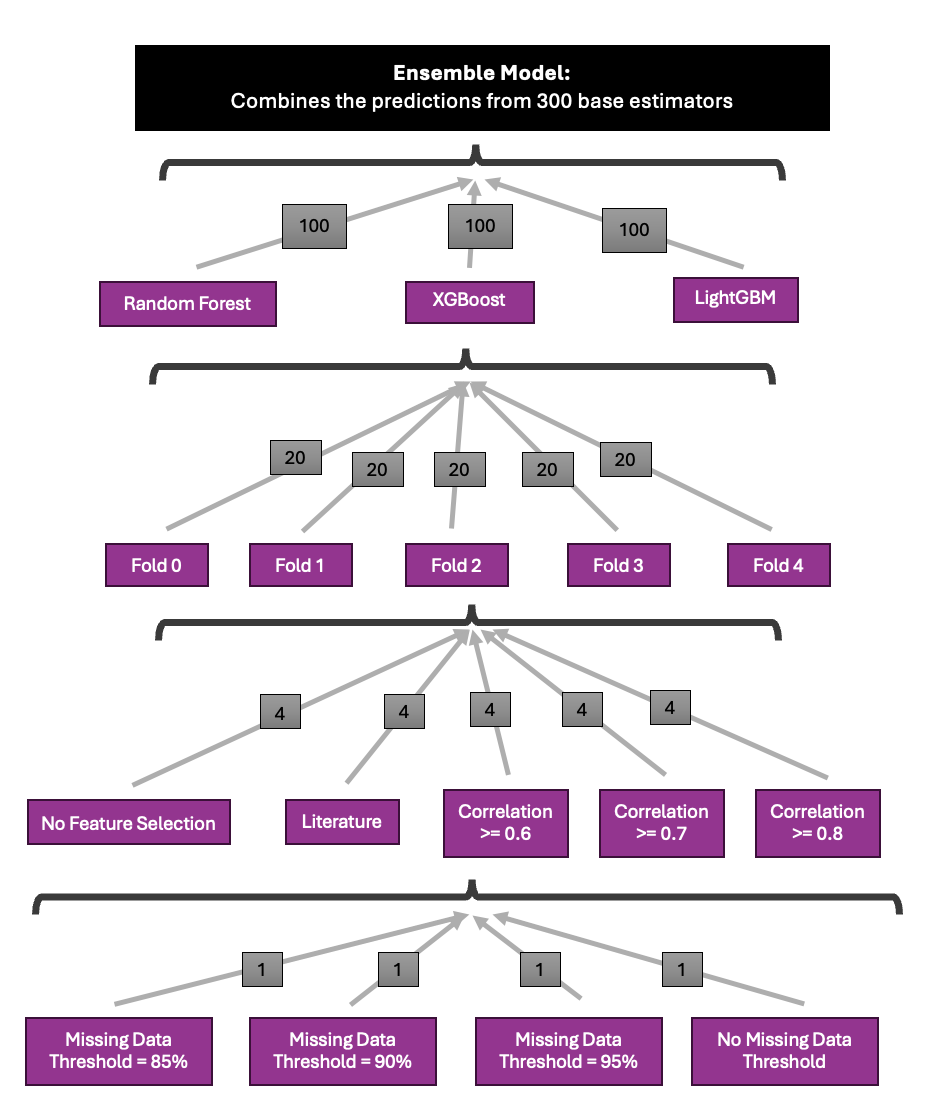
Each combination of model type, feature selection strategy and missing data removal strategy had 5 associated training datasets. The performance of the strategy combination was the average test performance of the models trained on these folds. The performances of different strategy combinations were then compared.

The 300 fine-tuned models for predictive data analysis were similarly tested, evaluated, and compared.

#### 4.413 Feature Importance Analysis

One of the primary aims of this research was to identify the socio-economic and health-related variables that have the highest predictive power for MMR. Using in-built methods implemented by the models’ platform, a feature’s importance can be calculated using the mean decrease in impurity/MSE across splits the feature is involved in (Scikit Learn Random Forest Regressor) or mean decrease in loss across the feature’s splits taking into account information from the loss function’s derivatives (XGBoost). Feature importance in LightGBM is calculated in a similar to way to feature importance in XGBoost, but LightGBM uses the total decrease in loss rather than the average decrease [16, 17]. These feature importances were used to provide information about the variables with the highest predictive power for MMR [18]. Additionally, variation in feature importance across the different models was used to investigate differences in the models’ performance.

### 4.42 Ensemble Models Training and Fine-Tuning



**Figure 9:** Visualisation of the source of the 300 base estimators used as input into the ensemble model. The purple boxes give the pre-processing method used to generate a version of the train dataset or the base estimator model used to produce an MMR estimate. The grey boxes give the number of base estimators trained per method. For example, the 300 base estimator predictions consisted of 100 predictions each from Random Forest, XGBoost, and LightGBM models. 20 model-specific predictions were made on each cross-validation fold, with these predictions consisting of 4 estimates for each feature selection method, each of which came from a dataset with a different missing data removal threshold.

#### 4.421 Using all Base Predictors in Different Types of Ensemble Models

Having hundreds of fine-tuned models creates the opportunity to use ensemble methods to combine predictions from multiple models, as described in Section 3.325. The ensemble modelling process detailed below was performed for both the missing data analysis and predictive analysis.

Predictions from 300 fine-tuned models were used as input data into an ensemble model, which learned how to best combine their predictions to produce an accurate, final MMR predictions. For the remainder of the thesis, the component fine-tuned models used as inputs to the ensemble will be referred to as “base estimators”.

More specifically, the base estimators were fit on their associated training data using their best hyperparameter settings, which were determined through the fine-tuning process described above. Then, each of the trained, fine-tuned models predicted the MMR for the the full, concatenated training and validation sets. In other words, the trained, fine-tuned models predicted on the training data before it was exposed to cross-fold validation, feature selection, and missing data removal. These predictions served as the training dataset for the ensemble models.

I used voting and stacking ensemble models to combine the base estimators’ predictions, as a review of ensemble methods given in Section 3.31 found that, of all models tested within a study, voting and stacking ensembles frequently produced the highest performance [20].

#### 4.4211 Voting Ensemble

For each country, year sample, the voting ensemble model produced a weighted average of the base estimators’ predictions. The weighting given to each base model was determined through 1,000 Optuna fine-tuning trials, with weights ranging from 0 to 1. The weightings used in the voting ensemble that had the lowest MSE on the train dataset were chosen as the optimal hyperparameters. No validation set was used because the voting ensemble was not ‘trained’, it was simply given different sets of weights to use to combine the various base estimators. Thus, the ‘training data’ served the same function as ‘out of sample validation data’.

#### 4.4212 Stacking Ensemble

The stacking ensemble method used a meta-learner that took predictions from the 300 base estimators as input and learned how to combine them to produce the lowest predictive error. Given that stacking ensemble models had internal parameters to tune, rather than simply hyperparameters like in the case of the voting ensemble, the training dataset had to be split into train and validation sets. This allowed the stacking ensemble models to fit their internal parameters on the train set and fine-tune their hyperparameters on the validation set over 1,000 Optuna trials. The ensemble training data was split into train/validation sets in an 80:20 ratio using Scikit Learn’s ‘train\_test\_split’ method [21]. The set of hyperparameter values that produced the lowest MSE on the validation set were used in the final, fine-tuned ensemble models.

I created three versions of the stacking ensemble to compare the performance of using the Elastic Net linear regression model, Random Forest regressor, and Support Vector Regressor as the stacking meta-learners. All models were used with the Scikit Learn’s implementation [22, 16, 23]. Elastic Net was used as a progression of the voting ensemble model, where the combination of L1 and L2 regression can both perform feature selection and reduce the possibility of overfitting, as described in the background [24]. Additionally, as described in the literature review, decision-tree based stacking ensembles outperform base estimators solely based on bagging and boosting, thus motivating use of the Random Forest regressor as a meta-learner [26]. Support vector regression was used as a meta-learner because its approach of only using datapoints outside its error tolerance margin could have interesting effects on how it uses the predictions from different base learners [25].

**Table 10:** Hyperparameter Tuning for Stacking Ensemble Models

|  |  |  |  |
| --- | --- | --- | --- |
| **Ensemble Model** | **Hyperparameter Name** | **Hyperparameter Function** | **Range of Potential Values** |
| Elastic Net Stacking Ensemble | Alpha | Specifies the overall extent of regularisation. | 0.1 to 1 |
| L1\_ratio | Controls the weighting of the L1 norm versus the L2 norm. High value makes the regularisation term closer to the L1 norm. | 0 to 1 |
| Random Forest Stacking Ensemble | Same parameters are described in Table 7. | | |
| Support Vector Machine Stacking Ensemble | kernel | Type of kernel used to transform input into a non-linear space. If ‘poly’, degrees tested were 2-5. | ‘poly’ (polynomial) or ‘rbf’ (radial basis function) |
| C | Strength of regularisation term | 0.1 to 1 |
| epsilon | Error tolerance, used to determine support vectors and decide which support vectors to include in loss. | 0.05 to 1 |

#### 4.422 Evaluating the Ensemble Models

A test set was generated to be able to compare the ensemble models’ predictive performance on out-of-sample data. The ensembles’ input test data consisted of the base estimators’ predictions on their test sets. The ensemble models then used these predictions to estimate MMR values, which were compared to the test ground truth. The ensemble models’ predictive performance on this test set was used to assess their generalisability and determine the best ensemble model.

#### 4.423 Using the Best Performing Ensemble to Test Different Combinations of Base Estimators

After establishing the best performing type of ensemble model, I tested whether I could improve the ensemble’s predictive performance by using different combinations of base estimators. More specifically, I determined which of the following combination of base estimators produced the highest predictive performance when combined by the previously established best performing ensemble:

* 300 base estimators consisting of XGBoost, LightGBM, and Random Forest regressors.
* 100 base estimators consisting of just XGBoost regressors.
* 100 base estimators consisting of just LightGBM regressors.
* 100 base estimators consisting of just Random Forest regressors.

The predictive performance of the different versions of the ensemble model was evaluated and compared using the same method as described in Section 4.422.

This investigation was made for both the missing data and predictive performance analyses.

#### 4.424 Analysis of Base Estimator Importance

To better understand the variation in the ensemble models’ predictive performance, I explored how each ensemble valued the contribution of the various base estimators. More specifically, I investigated whether different ensemble models placed the most importance on predictions from the same set of base estimators or if they placed the most importance on different base estimators.

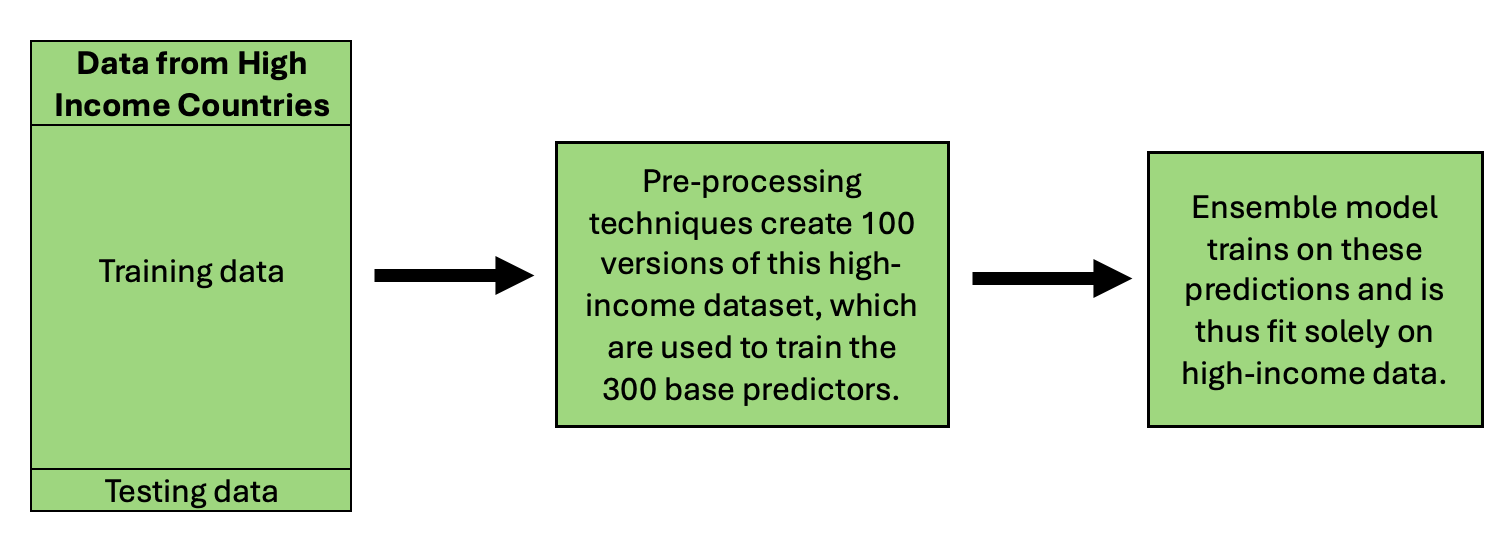
Since the base estimators’ predictions are the input to the ensemble models, the different base estimators can be thought of as “features”. Thus, the importance placed on each base estimator by the Random Forest ensemble was quantified using the same Scikit Learn feature importance calculation described in Section 4.413 [16]. In contrast, the weighting of each base estimator in the voting ensemble’s final prediction was used as a proxy for model importance. Similarly, base estimator importance in the Elastic Net model was determined using the coefficient attached to each base estimator’s predictions.

#### 4.425 Uncertainty Analysis

I explored the variation in base estimators’ predictions for the same sample to provide a measure of uncertainty in the predictions of the best performing ensemble. More specifically, there were 300 base estimator predictions for each ground truth MMR estimate. I calculated the standard deviation among the 300 predictions for every datapoint in the test set to explore the base estimators’ certainty in the true MMR prediction. Lack of consensus among the base predictors would likely make the ensemble’s prediction less stable. As a result, this analysis provides an approximation for uncertainty in the ensemble’s predictions.

## 4.5 Sensitivity Analysis

I conducted a sensitivity analysis to gain a deeper understanding of how the input dataset affected the quality of the best performing ensemble’s predictions. The sensitivity analysis was conducted using the same procedure for the missing data analysis and predictive analysis models. See Figure 9 for a summary for the sensitivity analysis procedure.



**Figure 10:** Visualisation of the sensitivity analysis procedure, where the predictions used to train the ensemble model were only fit on high-income data. As described in Section 4.42, the ensemble was trained on the predictions from the 300 base estimators, which were trained on different versions of the high-income data. The sensitivity analysis created a separate ensemble model for each income-specific version of the input dataset.

To perform the sensitivity analysis, I created 4 new versions of each fold/feature subset datasets by filtering the input dataset by income level. I then conducted missing data thresholding on each filtered dataset, as above. The filtered datasets will be referred to as “sensitivity datasets” from this point forward, with the four filtered datasets characterised in the following list. This process generated 300 base estimators trained on each sensitivity dataset.

* Data from low-income countries only.
* Data from lower-middle income countries only.
* Data from upper-middle income countries only.
* Data from high-income countries only.

A separate ensemble model was created for each set of 300 base estimators using the architecture and hyperparameters of the best performing ensemble model. The training data for each sensitivity analysis ensemble consisted of the trained base estimators’ predictions on a filtered version of the concatenated train, validation data, which only contained data from the relevant income level. The remainder of process followed the same procedure described in Section 4.411.

The sensitivity analysis ensemble models were then evaluated using a version of the test dataset filtered to only contain data from the relevant income level. Each sensitivity analysis ensemble was compared to the original best performing ensemble, which was trained on data from all income levels but only used to predict on the filtered test set. For example, the sensitivity analysis ensemble trained and tested on data from low-income countries was compared to the original ensemble trained on all data but tested on data from only low-income countries. Using the same test set ensured comparable performance between the original and sensitivity analysis models.

## 4.6 Comparison to Literature

I compared my best performing ensemble, trained on data from all income levels, to the latest versions of the UN MMEIG’s BMat model, GBD Study’s CODEm model, and the GMatH simulation model. More specifically, I explored the differences between my model’s MMR predictions for each country/year sample to the predictions made by models described in the literature [27, 28, 29]. The MMR estimates from the literature were given with their 95% confidence intervals.

First, I calculated the percentage difference between my test set MMR estimates and the corresponding estimates from each literature model using Equation 12, below. This gave an indication of the similarity between my estimates and the literature’s predictions.

Then, I determined the percentage of test set MMR estimates from my best performing ensemble that fell in the 95% confidence interval of the corresponding estimates from the literature. This analysis was performed for all test set estimates and per income level, where I calculated the proportion of test set estimates for countries from a particular income that fell within the 95% confidence intervals of the associated estimates from the literature. I also calculated the proportion of ground truth MMR estimates used to test my model that fell within the 95% confidence interval of the literature models’ estimates. This compared the MMR estimates that my model was trained to predict with the literature’s MMR estimates to provide more information about whether differences between my model and the literature were due to poor performance or training data.

Finally, I visualised the difference between my model’s MMR estimates and the literature’s estimates for an exemplar country from each income level. I visualised the model’s estimates from the train set and test set separately to explore any possible difference. I attempted to maximise geographic coverage by comparing estimates for at least one country in the Americas, Africa, Europe, Oceania, and Asia.

Citations

1. Onambele, L.; Guillen-Aguinaga, S.; Guillen-Aguinaga, L.; Ortega-Leon, W.; Montejo, R.; Alas-Brun, R.; Aguinaga-Ontoso, E.; Aguinaga-Ontoso, I.; Guillen-Grima, F. Trends, Projections, and Regional Disparities of Maternal Mortality in Africa (1990–2030): An ARIMA Forecasting Approach. Epidemiologia 2023, 4, 322–351. https://doi.org/ 10.3390/epidemiologia4030032
2. <https://link.springer.com/article/10.1186/s40537-021-00516-9>
   1. Journal of big data
   2. Missing data mechanisms
3. <https://link.springer.com/article/10.1007/s40273-023-01297-0#Sec2>
   1. More about missing data mechanisms
4. <https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html>
5. <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>
6. <https://www.nature.com/articles/s41592-018-0019-x>
   1. Curse of dimensionality
7. <https://jmai.amegroups.org/article/view/8590/html>
8. <https://www.sciencedirect.com/science/article/pii/S0169023X23000587?pes=vor&utm_source=scopus&getft_integrator=scopus>
9. <https://iris.who.int/bitstream/handle/10665/381012/9789240108462-eng.pdf?sequence=1>
10. <https://onlinelibrary.wiley.com/doi/10.5694/mja2.52452>
    1. Non-communicable disease and pregnancy
    2. Ramson et al., 2024
11. Akselrod S, Banerjee A, Collins TE, Acharya S, Artykova N, Askew I, Berdzuli N, Diorditsa S, Eggers R, Farrington J, Jakab Z, Ferreira-Borges C, Mikkelsen B, Azzopardi-Muscat N, Olsavszky V, Park K, Sobel H, Tran H, Vujnovic M, Weber M, Were W, Yaqub N, Berlina D, Dunlop CL and Allen LN (2023) Integrating maternal, newborn, child health and non-communicable disease care in the sustainable development goal era. Front. Public Health 11:1183712. doi: 10.3389/fpubh.2023.1183712
12. [**https://pubmed.ncbi.nlm.nih.gov/27642018/**](https://pubmed.ncbi.nlm.nih.gov/27642018/)
13. [**https://pmc.ncbi.nlm.nih.gov/articles/PMC11117177/pdf/jogh-14-04128.pdf**](https://pmc.ncbi.nlm.nih.gov/articles/PMC11117177/pdf/jogh-14-04128.pdf)
14. [**https://www.sciencedirect.com/science/article/pii/S2589537024002323**](https://www.sciencedirect.com/science/article/pii/S2589537024002323)
15. <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GroupKFold.html>
    1. K-fold cross validation mechanism
16. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html#sklearn.ensemble.RandomForestRegressor>
    1. Scikit learn random forest
17. <https://xgboost.readthedocs.io/en/stable/parameter.html>
    1. Xgboost
18. <https://lightgbm.readthedocs.io/en/latest/Parameters.html>
    1. Lightgbm
19. [**https://dl.acm.org/doi/10.1145/3292500.3330701**](https://dl.acm.org/doi/10.1145/3292500.3330701)
    1. Optuna
20. <https://pmc.ncbi.nlm.nih.gov/articles/PMC10298658/>
    1. Ensemble methods review
21. <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html>
    1. Train test split method
22. <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html>
    1. Elastic net
23. <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html#sklearn.svm.SVR>
    1. Support vector regressor
24. <https://academic.oup.com/jrsssb/article/67/2/301/7109482>
    1. Elastic net
25. <file:///Users/rosalitarosenberg/Downloads/B_STCO.0000035301.49549.88.pdf>
    1. A tutorial on support vector regression
    2. Smola and Sch¨olkopf
26. <https://www.nature.com/articles/s41598-024-71934-x#Abs1>
    1. Ensemble article with high performance from DT based methods
27. <https://www.who.int/data/gho/data/indicators/indicator-details/GHO/maternal-mortality-ratio-(per-100-000-live-births)>
    1. UN MMEIG data
28. Global Burden of Disease Collaborative Network.
    1. Global Burden of Disease Study 2021 (GBD 2021) Results.
    2. Seattle, United States: Institute for Health Metrics and Evaluation (IHME), 2022.
    3. Available from <https://vizhub.healthdata.org/gbd-results/>.
29. Ward, Zachary, 2023, "Simulation Results: Global, regional, and country-level maternal mortality by cause, 1990-2050", <https://doi.org/10.7910/DVN/UBGY9P>, Harvard Dataverse, V1, UNF:6:YbDwFpLTGF39vO7onL7qww== [fileUNF]
30. <https://genderdata.worldbank.org/en/indicator/sh-sta-mmrt?estimate=National>
    1. Mmr data
31. <https://genderdata.worldbank.org/en/topics/health#idAllIndicators>
    1. World bank data
32. <https://www.who.int/data/sets/health-inequality-monitor-dataset#swper>
    1. Empowerment data
33. <https://www.who.int/data/sets/health-inequality-monitor-dataset#wb>
    1. Health determinants data (mostly economic)
34. <https://www.who.int/data/sets/health-inequality-monitor-dataset#ihme-gbd>
    1. Illness prevalence data
35. <https://www.who.int/data/sets/health-inequality-monitor-dataset#gho>
    1. Income level data