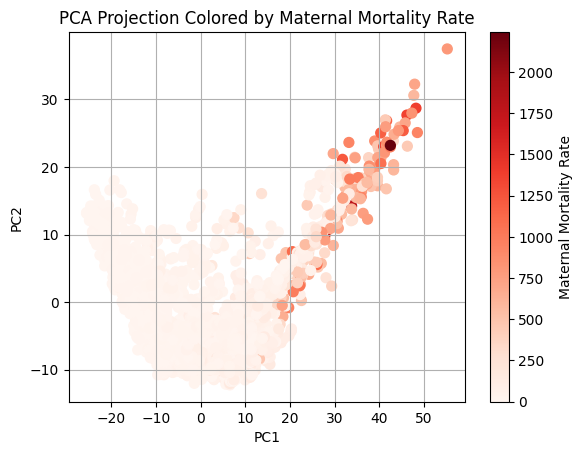
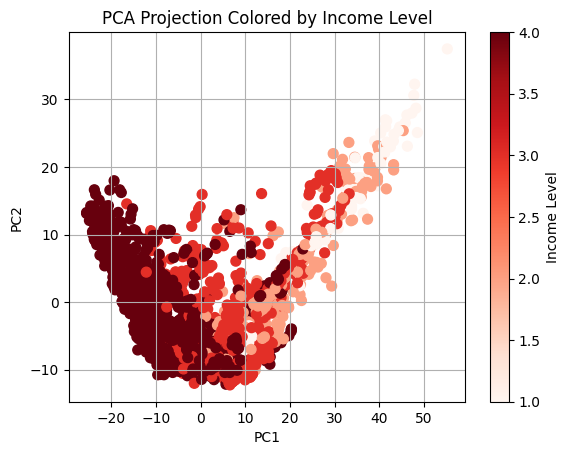
**Agenda**:

* Initial Data Exploration: PCA and key summary statistics
* Quantifying the variation across folds for models trained with all features
* Quick confirmation of hypothesis testing pipeline
* Experimental process flowchart
* Finetuning Strategies
  + Literature Search
  + By correlation strength
    - Initial variance exploration
  + By model designated importance

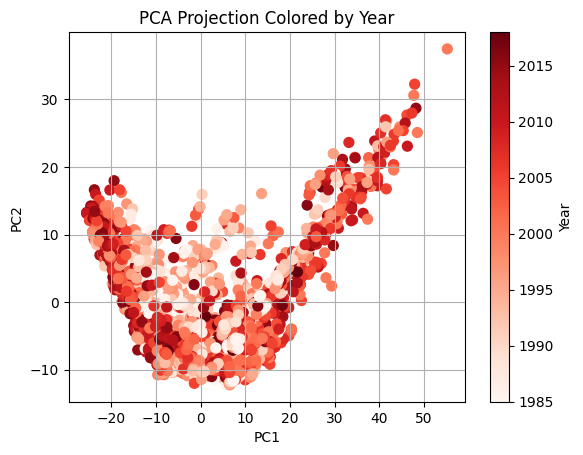
**Initial Data Exploration**

*PCA*

1. I took the merged data collected between 1985 and 2018, then dropped any columns with all missing data and rows where the MMR estimate was NAN.
2. I replaced all country names with a number encoding.
3. I imputed the rest of the missing data using KNN Imputation (as the PCA implementation could not work with missing data).
4. Standardised the data to allow best PCA performance to prevent variance calculations from being skewed by features with large magnitudes.
5. Application of Scikit Learn’s PCA tool.
6. Plotting PCA results with axes defined by the 1st and 2nd principal components.

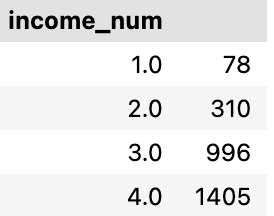
According to the plots, it appears that the cluster of data belonging to the highest income countries (darkest dots on the right) correspond to the cluster with the lowest maternal mortality rates (lighest dots on the right).



There did not appear to be obvious year-based clusters on the PCA graph.

*Key Summary Statistics:*

I took the merged data within 1985 and 2018 and used the Panda’s groupby functions to find the median, standard deviation and missing data proportion for several key features. Median was used instead of the mean to prevent the summary statistics from being biased by outliers. These features were a subset of the features that the literature indicated as the most influential for maternal mortality rates.



* The number of samples from each income level. The highest income level, had the most samples which could bias the model to higher income countries.
* The lack of data from lower income countries also motivates this study.

A screenshot of a computer

AI-generated content may be incorrect.

* As income level increased, maternal mortality rate decreased.
* Mortality rate standard deviation also decreased as income level increasing, indicating that the highest mortality rates span a greater range, and that countries with lower incomes have more varying health outcome landscapes.
* No missing data (because it was removed during data pre-processing).

A screenshot of a graph

AI-generated content may be incorrect.

* As income level increased, the proportion of females who reached at least 65 years old also increased.
* The largest increase in survival occurred between the lowest and second lowest income countries, with change between income levels staying between 7 and 8 percent for the other transitions.
* There was no missing data for this metric.

A screenshot of a cell phone

AI-generated content may be incorrect.

* The percentage of women living with HIV decreased as income level increased, with all the lowest income countries reporting this metric but with 15-17% of data for this metric missing for higher income countries.
* Standard deviation was similar across income levels.

A screenshot of a computer

AI-generated content may be incorrect.

* As income level increased, the proportion of women participating in their own health care decisions also increased (with standard deviation decreasing).
* However, the share of missing data in this metric increased with income level, potentially because it was not measured in many high-income countries.
  + The lack of data prevents generalisation to all high-income countries.

A screenshot of a graph

AI-generated content may be incorrect.

* As income level increased, infant mortality rate decreased, with very little missing data in this metric.

A screenshot of a graph

AI-generated content may be incorrect.

* The percentage of pregnant women receiving prenatal care increased as income level increased.
* However, the missing data proportion in this metric increased with income level, again making it difficult to generalise the findings to all high-income countries.

A screenshot of a graph

AI-generated content may be incorrect.

* Similarly, as income level increased, the proportion of births attended by skilled health staff increased, and standard deviation decreased notably.

A screenshot of a graph

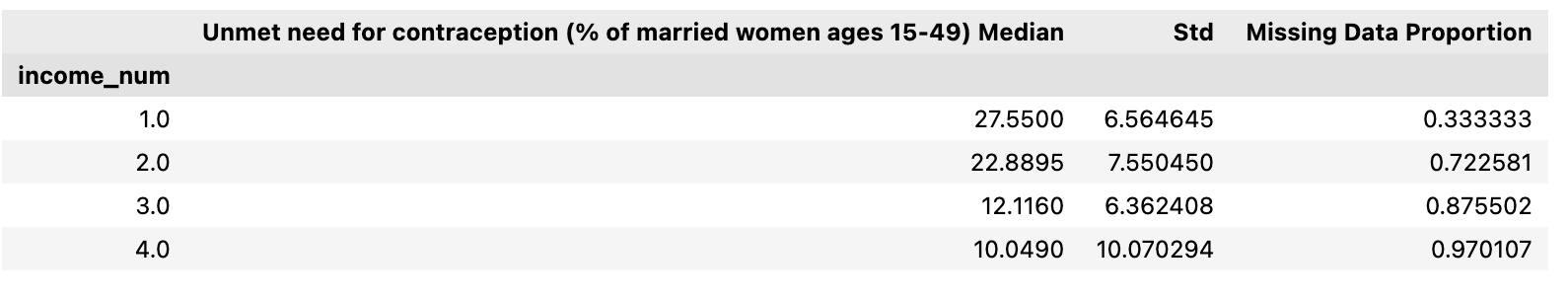
AI-generated content may be incorrect.

* Stillbirth rate decreased as income level increased.

A screenshot of a white and black text

AI-generated content may be incorrect.

* As income level increased, the prevalence of communicable, maternal, neonatal, and nutritional diseases per 100,000 females in the population decreased.
* Standard deviation did not change uniformly in one direction as income level changed.
* Missing data proportion was high for all income levels



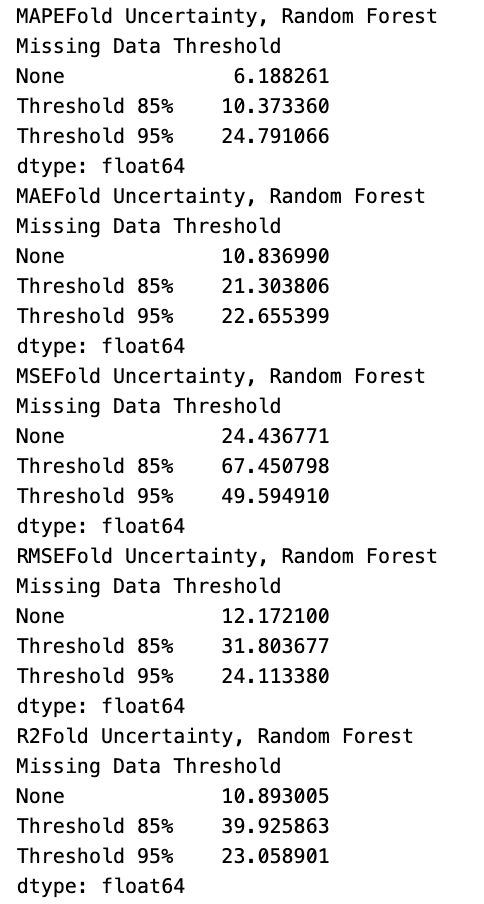
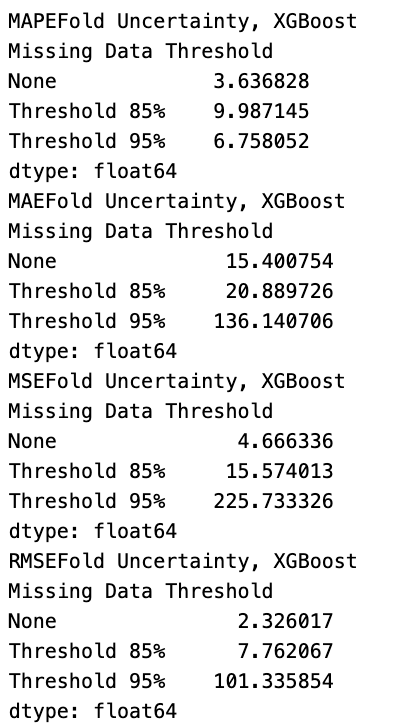
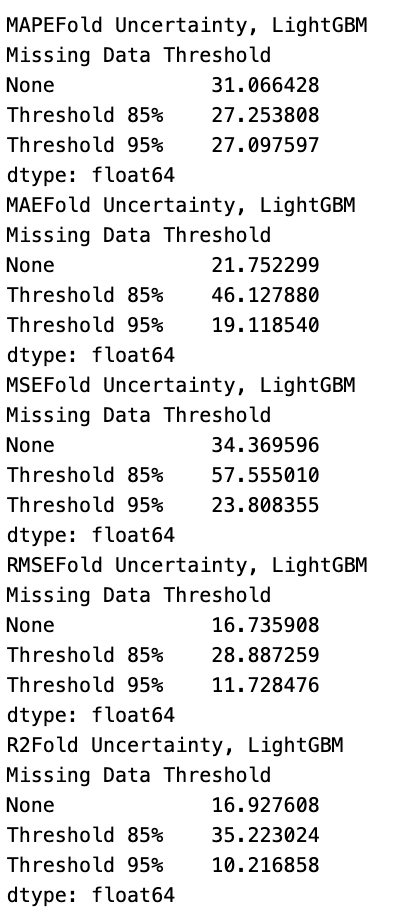
* This metric had more missing data for the higher-income countries, making it difficult to generalise any trends in those higher-income countries.
* However, generally, there was a smaller unmet need for contraception in higher income countries.

**Uncertainty Across Folds**:

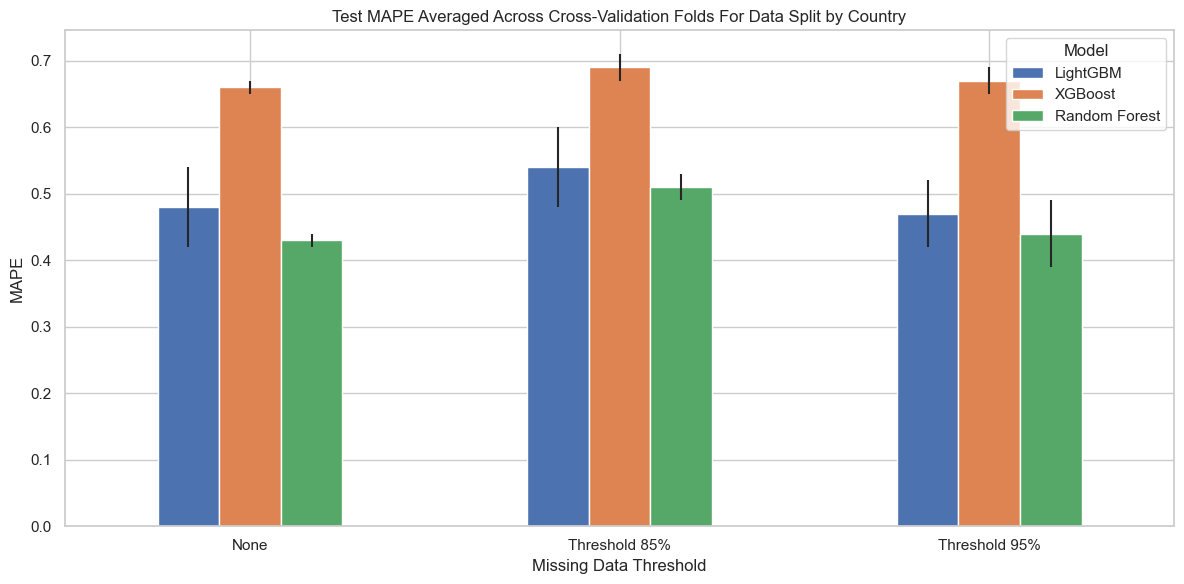
To better understand the difference across folds, I divided the difference between the maximum and minimum metrics by the average metric across the folds for each model/missing data threshold combination. This was only done for models not trained on a subset of features.

**Value = (max metric across folds – min metric across folds)/(average across folds) \*100%**

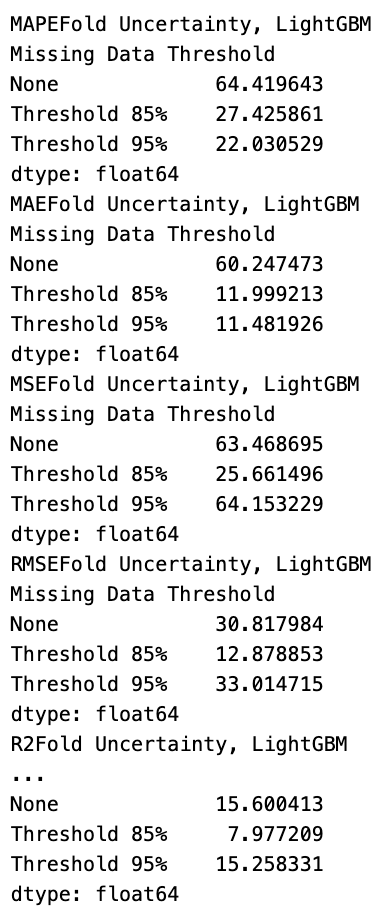
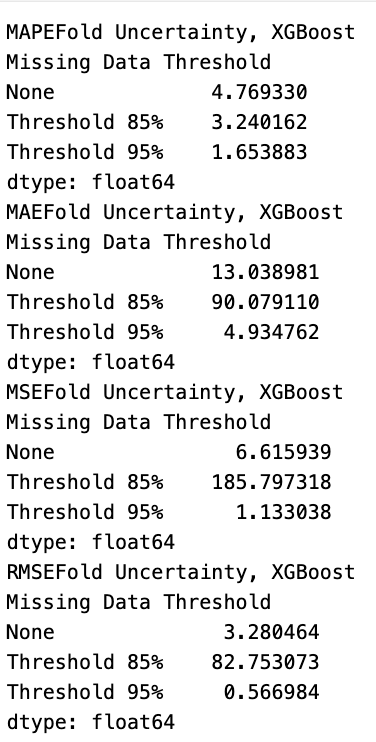
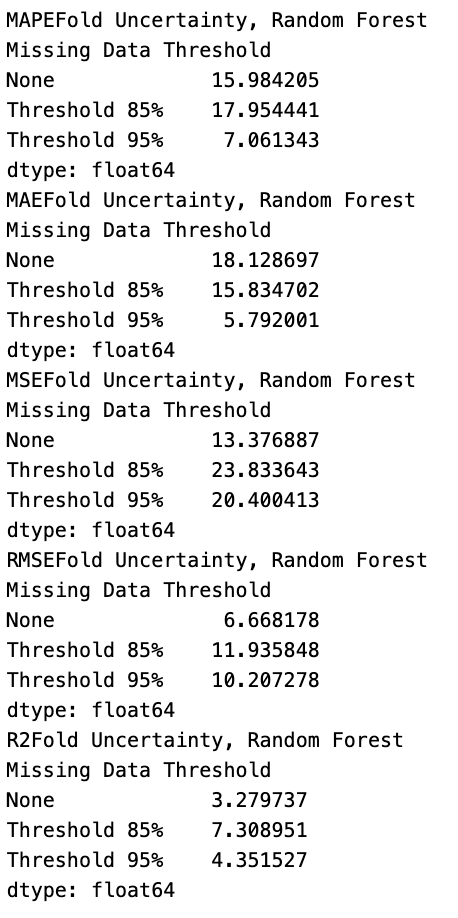
*By Country*:



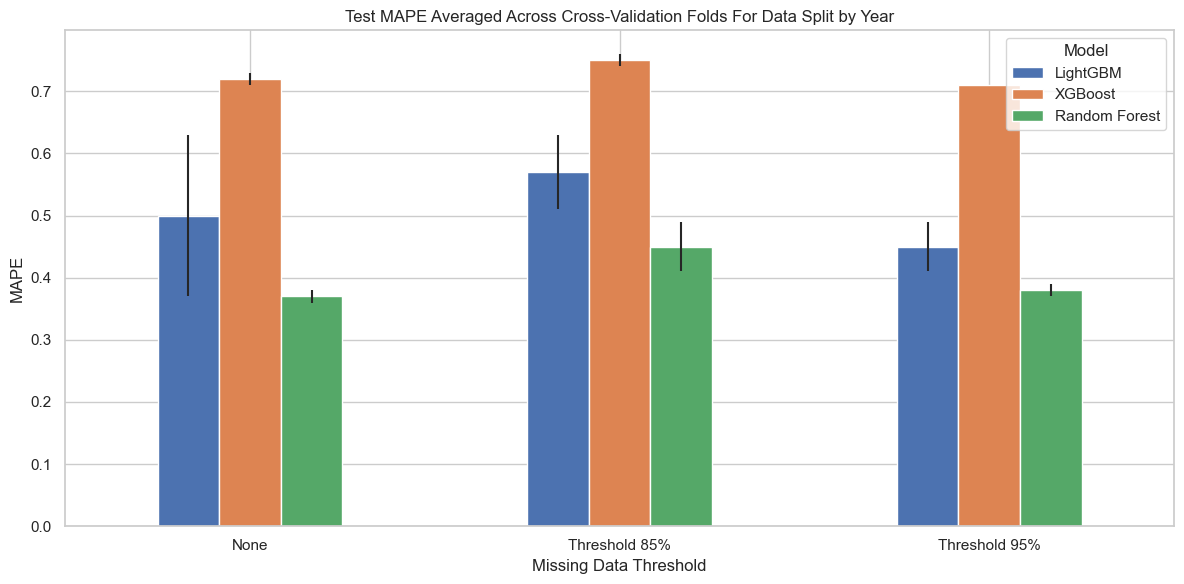
* LightGBM models had the highest uncertainty across folds for the threshold 85% data (except for relative error), potentially because features could vary more between folds.
* The same observation as above could be made about the Random Forest models , except that the fold-variation in MAE score was highest for the 95% threshold.
* The XGBoost models had high uncertainty across the threshold 95% data in all metrics other than the relative error.



*By Year*:

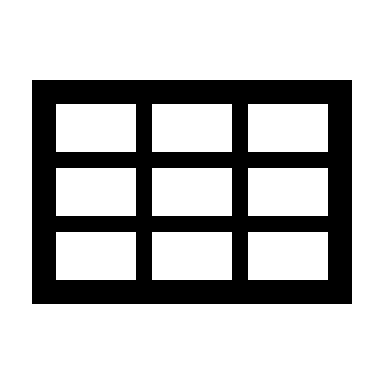
  

* LightGBM almost always had the highest intra-fold variation when no missing data was removed. It had the highest variation for this threshold across all models.
* Variation in XGBoost error scores was very high, especially in the MSE metric.
  + This could imply XGBoost is more affected by outliers than the other models.
* Random Forest had the highest variation across the 85% missing data threshold folds.



**Feature Selection: Experimental Workflow**

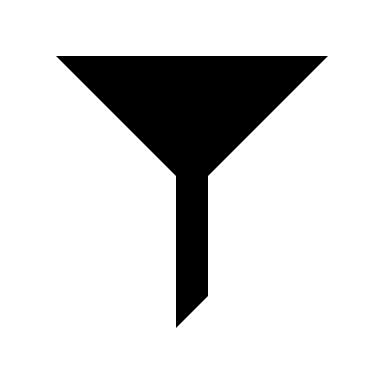
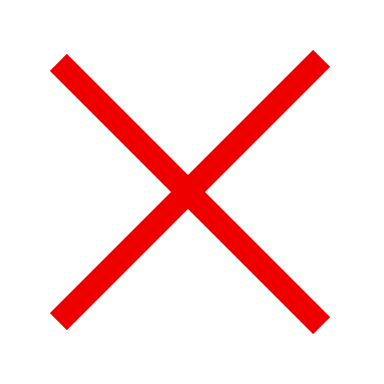
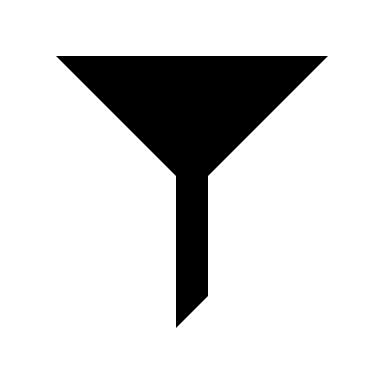
1. Take the testing set and cross-validation folds from the original dataset where no missing data removal or feature selection was performed and filter for the desired features.
2. Now remove missing data, creating three versions of each fold.
   1. Removed missing data after feature subsetting because the number of columns influences the proportion of missing data per row, affecting the iterative thresholding operation.
3. Train models on these new features, finetuning hyperparameters using 300 Optuna trials and MSE metric.
4. Save best hyperparameters for each fold and missing data threshold.
5. For testing, retrain model on fold training data with feature subset and best hyperparameters before calculating each accuracy metric based on test prediction accuracy.
6. Then compare to other model versions



Merged data

Feature selection

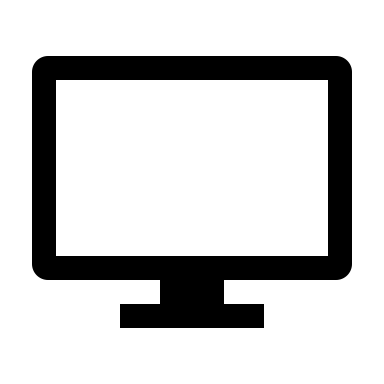
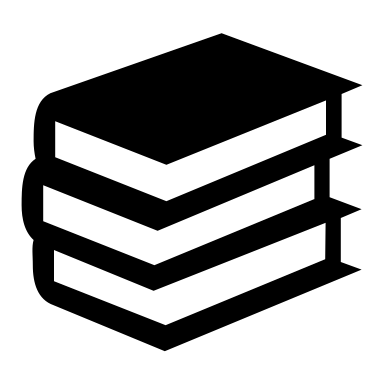
No feature selection

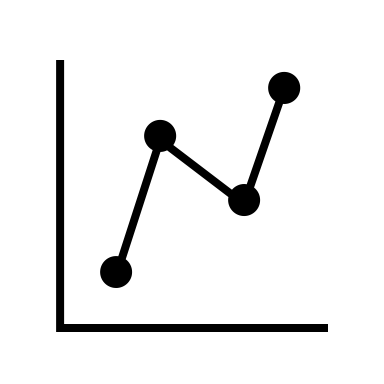
 

Model permutation importance

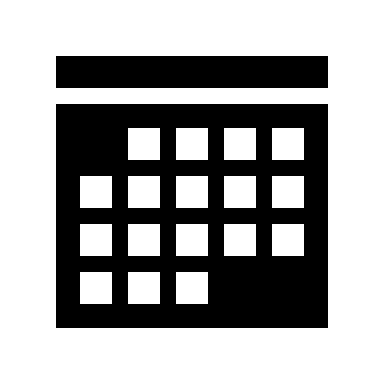
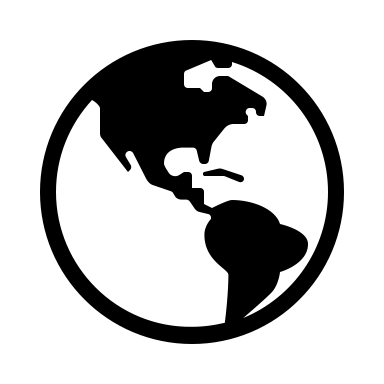
From literature

Correlation Strength



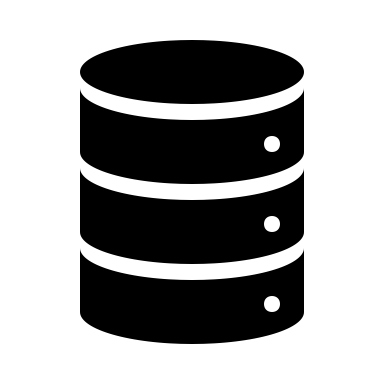


Incoming data for train/test split



Split by country

Split by year

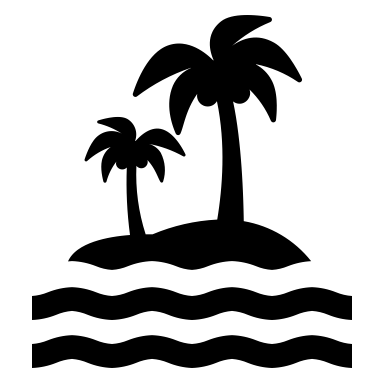
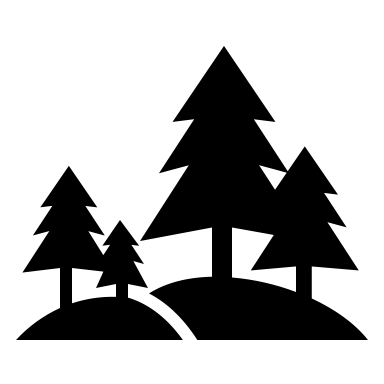
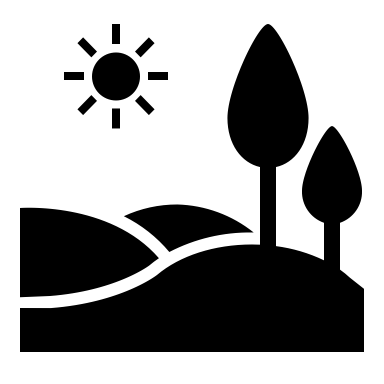


Splitting into cross-validation folds

85% missing data threshold

No missing data removal

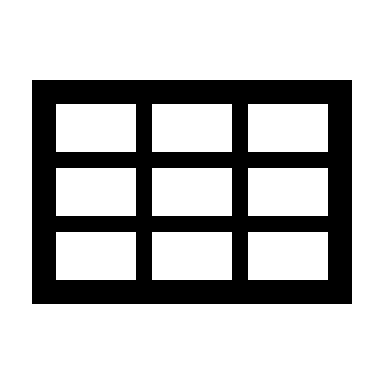
95% missing data threshold



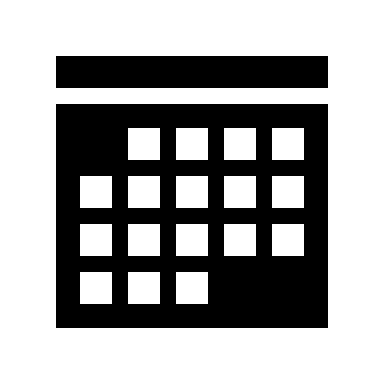
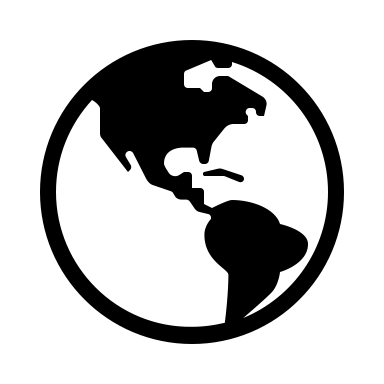
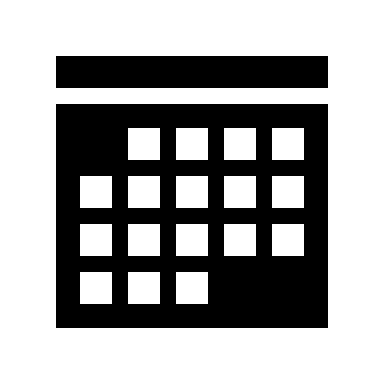
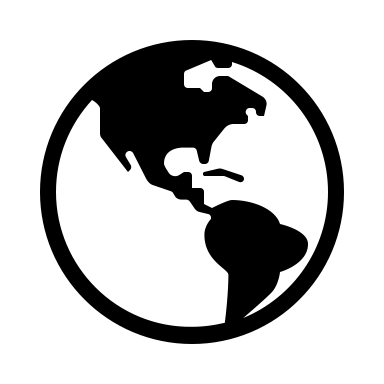
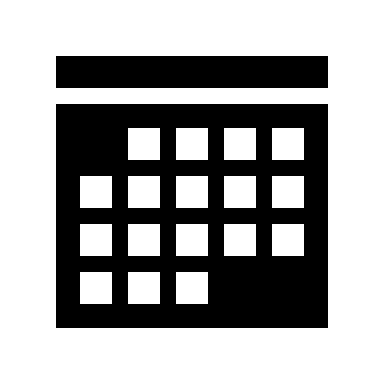
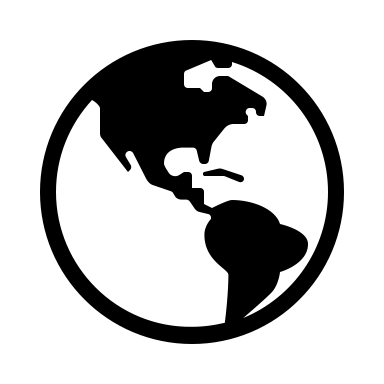
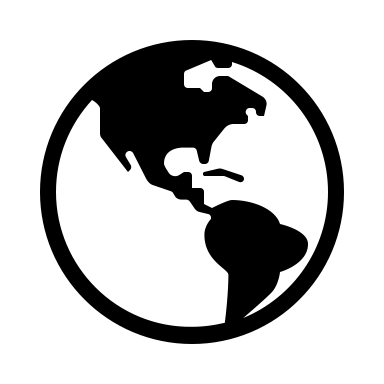
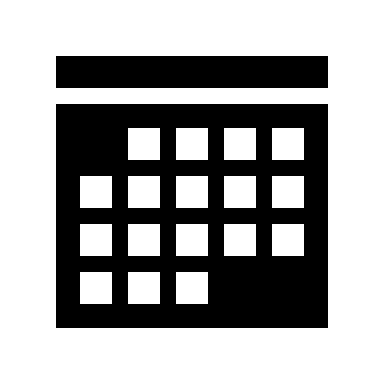
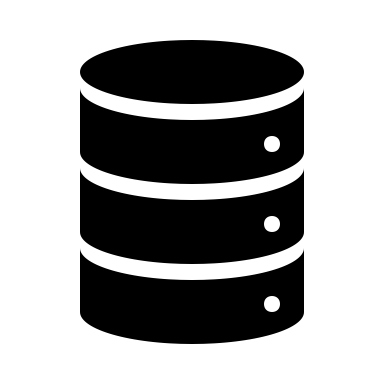
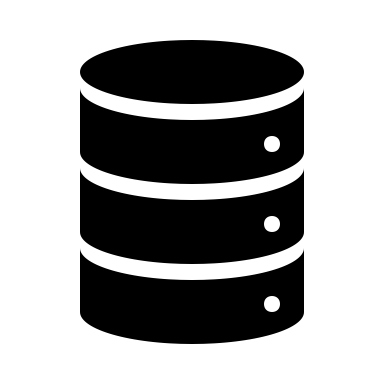
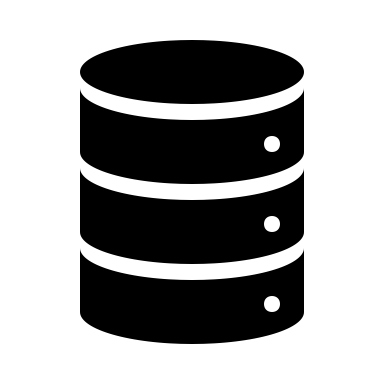
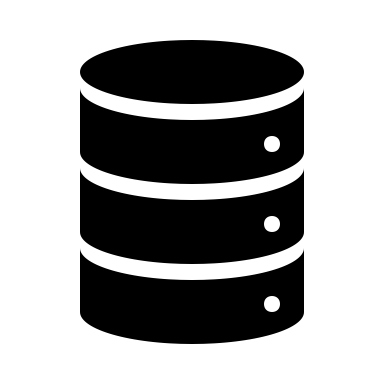
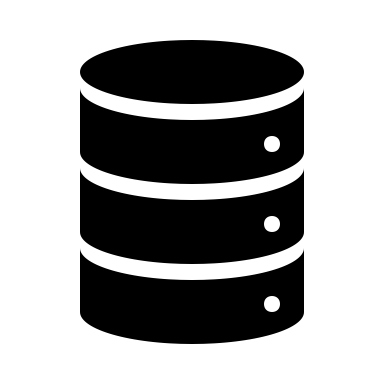
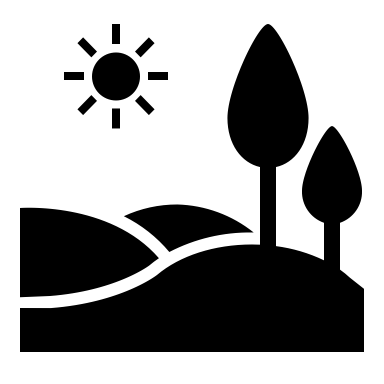
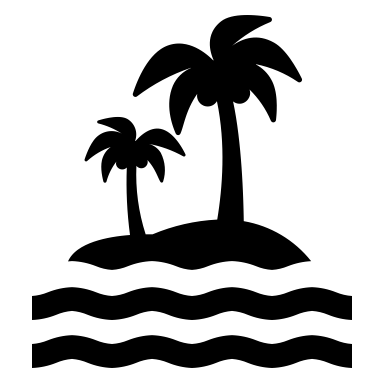
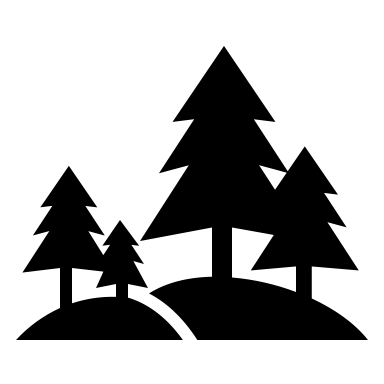
LightGBM

Random Forest

XGBoost



Incoming dataset



XGBoost

LightGBM

Random Forest

No missing data removal

85% missing data threshold

95% missing data threshold

Splitting into 5 cross-validation folds (80:20 train: validation)

Split into train/test data by year (90:10)

Split into train/test data by country (90:10)

**Feature Selection:**

All results will be presented after the description of methods for comparative purposes.

**Feature Selection Informed by the Literature**:

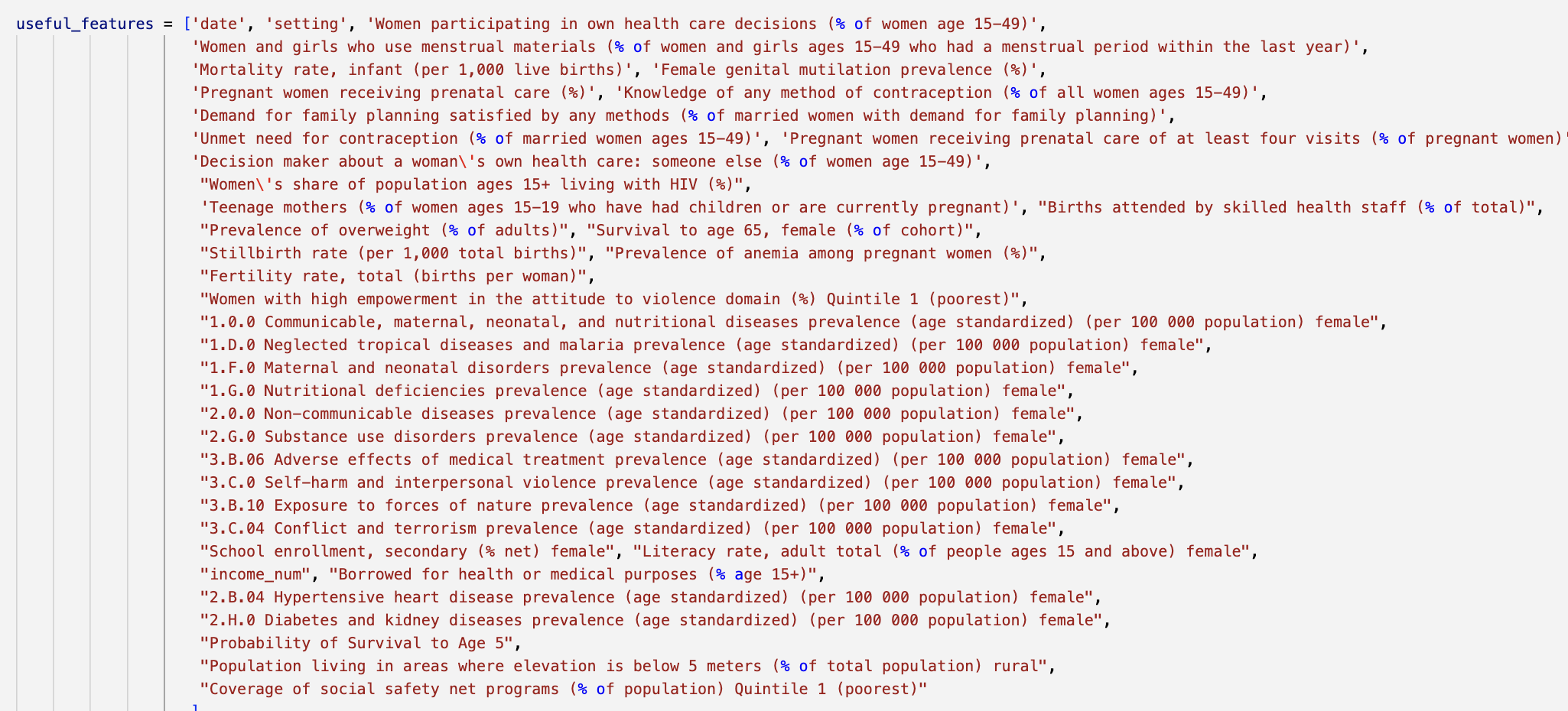
I surveyed a number of papers about maternal mortality to learn about which features researchers believed most strongly influence maternal mortality rates. I used the following papers and analysis from the WHO:

* <https://jmai.amegroups.org/article/view/8590/html>
* <https://www.sciencedirect.com/science/article/pii/S0169023X23000587?pes=vor&utm_source=scopus&getft_integrator=scopus>
* <https://iris.who.int/bitstream/handle/10665/381012/9789240108462-eng.pdf?sequence=1>

These papers indicated that a mixture of biological and socio-economic variables strongly influenced maternal mortality rates. Some of the most important were:

* The country’s life expectancy
* Pre-existing medical conditions
* Nutritional status
* High quality, accessible medical care throughout the pregnancy and during birth
* Lack of gendered bias in the medical system
* Affordable medical care
* Contraception
* Violence and environmental dangers
* C-section occurrence

As a result, I selected the following subset of 40 features to use to train my ML models.



While there were many more relevant features I could have chosen from the available dataset, I believed these covered the major maternal mortality determinants.

**Feature Selection Informed by Correlation Analysis**:

I computed the pairwise Pearson’s correlation coefficient of all feature columns with the national maternal mortality rate (MMR) estimate. I used the Pandas correlation method, which ignores rows where either feature pair has a missing value. Therefore, this method may behave strangely for columns with a lot of missing data, where the lack of data may make it look like there is either a false strong or weak correlation between the feature and MMR measurement.

From a literature search, it seemed like researchers were determining the best number of features experimentally, with methods used to cull the number of features to a subset between 20 and 200 features in length. Thus, I created three versions of the dataset, one for a subset of features with an absolute pairwise correlation coefficient greater than each of the following thresholds. I then trained models on each subset to compare the feature selection methods.

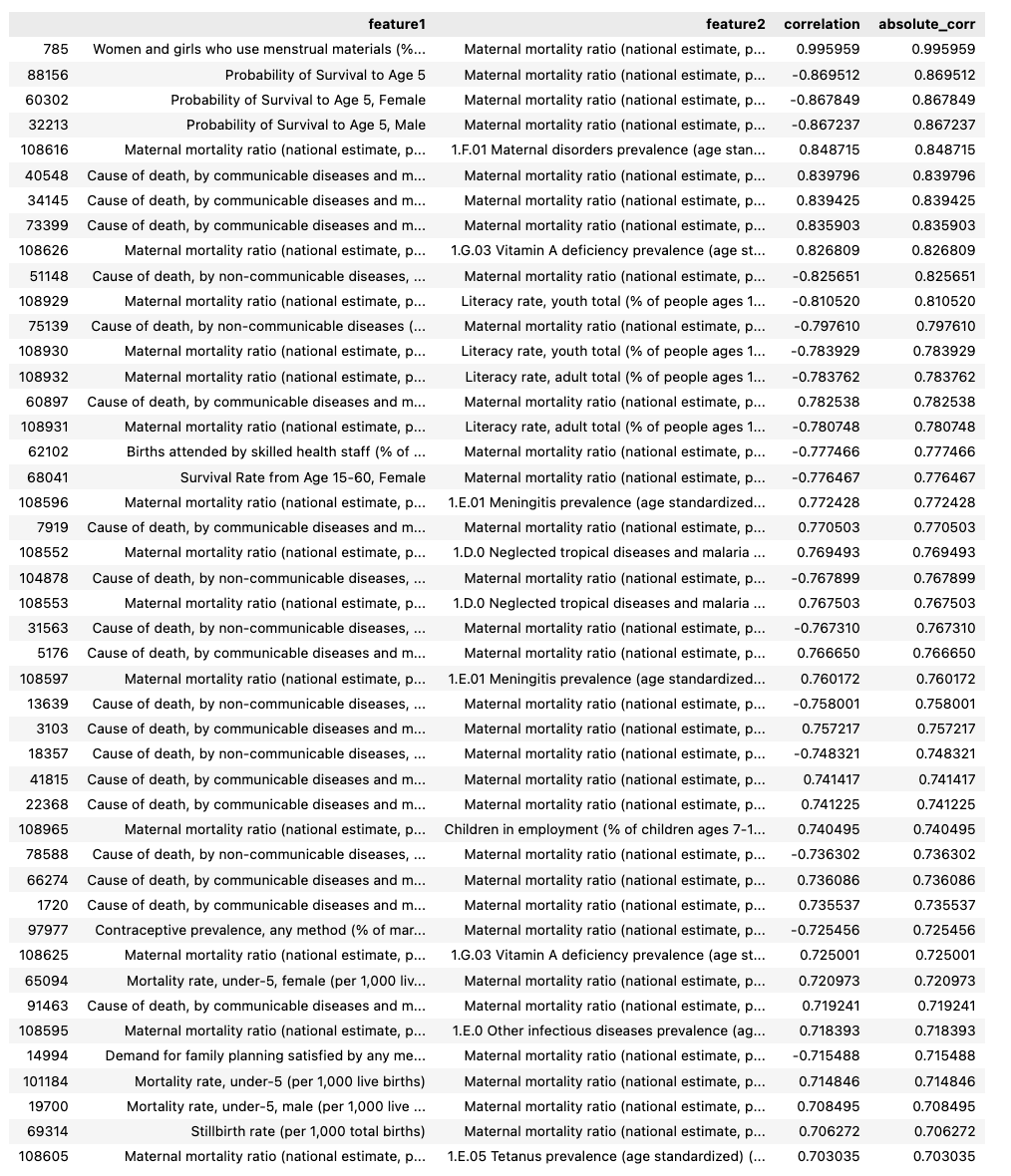
* Correlation threshold = 0.8, producing 11 features

A screenshot of a medical report

AI-generated content may be incorrect.

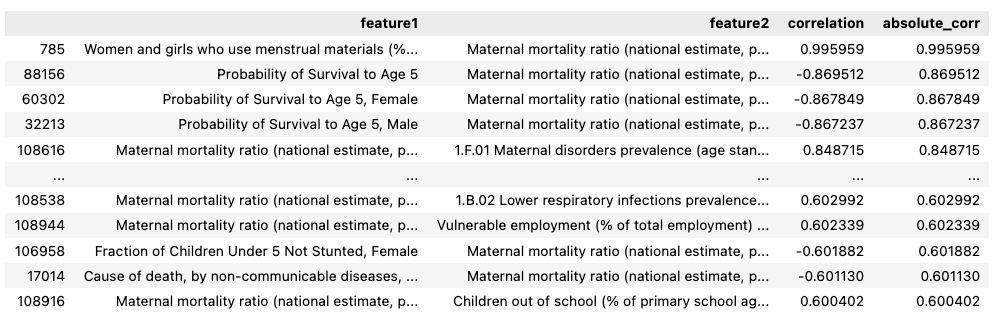
These chosen features are similar to those indicated by the literature as being important.

* Correlation threshold = 0.7, producing 45 features



Again, this covers many of the same features nominated by the literature as being important.

* Correlation threshold = 0.6, producing 113 features.

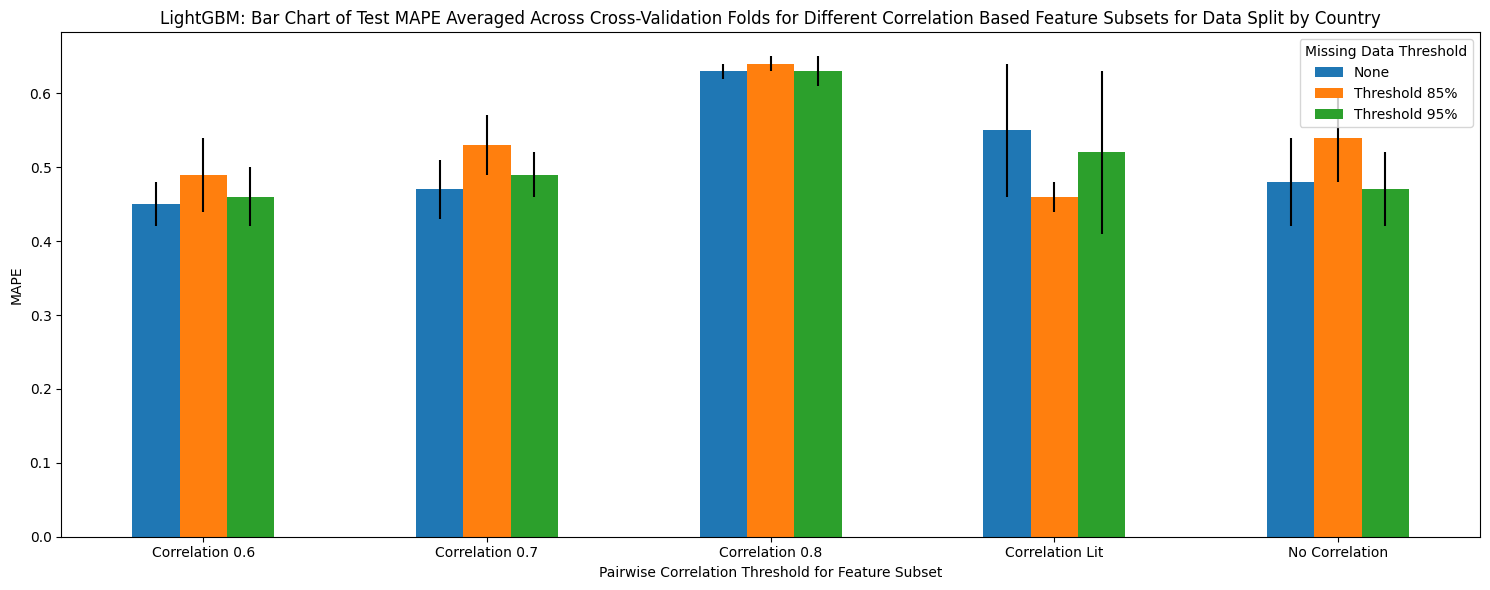


No columns were left for iterative threshold = 85%, correlation = 0.6, fold 0, data split by country (so results from this fold could not be gathered)

**Test Performance Comparison between Feature Subsets:**

I will be discussing the 3rd feature subsetting method below, as it was not implemented. I have plots of the relative performance across all the metrics, but for conciseness am only showing MAPE. I can show you the other plots if you would like, as there are slight differences between the best performing model version depending on the metric used.

*By Country*:



A graph of different colored bars

AI-generated content may be incorrect.

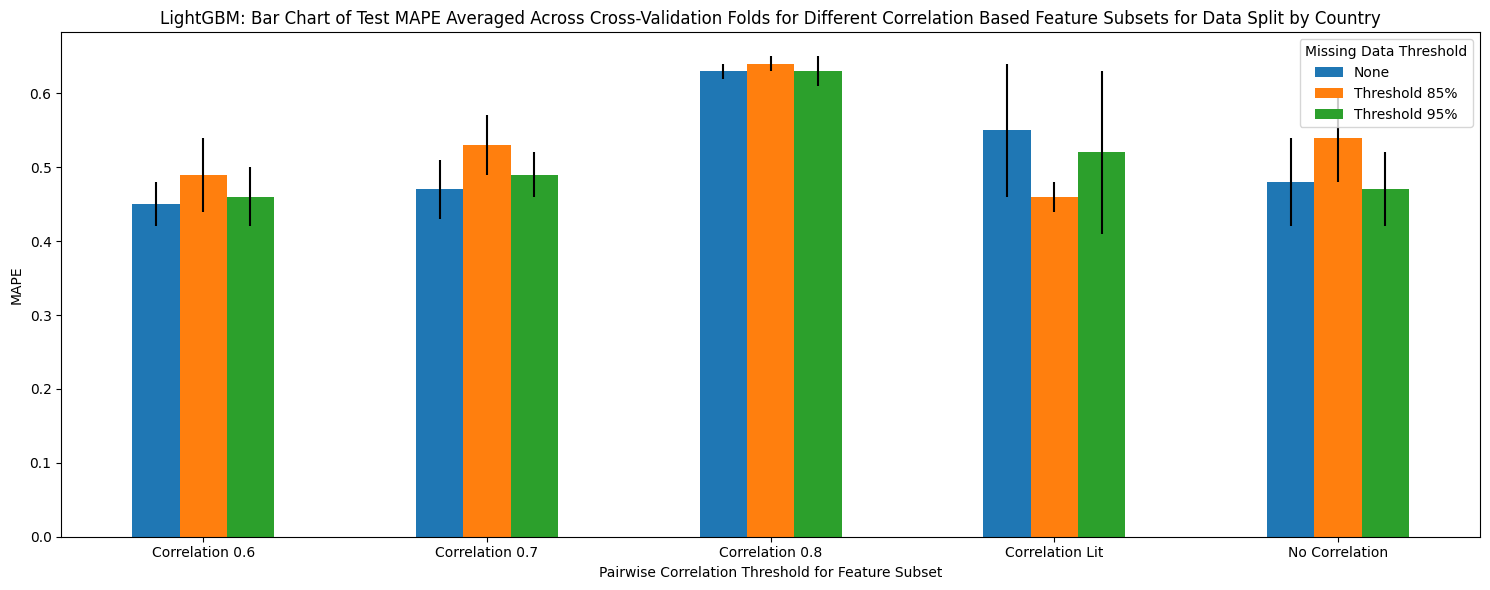
* The random forest and lightGBM models both had the worst performance on the correlation=0.8 subset, with feature selection not noticeably improving performance.
  + The lightGBM model performed best with feature selection (correlation=0.6) and no missing data removal.
  + The poor performance on the correlation=0.8 subset may have been due to the low number of features used for training and testing (=11).

A graph of different colored bars

AI-generated content may be incorrect.

* Relative error increased as correlation threshold increased, becoming largest when no correlation was performed.
* Relative error was lowest when the missing data threshold = 85% and correlation=0.6.

*By Year*:



A graph of different colored bars

AI-generated content may be incorrect.

* Taking a feature subset sometimes reduced variation across cross-validation folds (the error bars), potentially due to a lack of noise.
* The random forest and lightGBM models both had the worst performance on the correlation=0.8 subset, with no feature selection having similar performance to the correlation=0.6 threshold. While the random forest model performed slightly better with no feature selection or missing data removal, the lightGBM model performed best with feature selection (correlation=0.6) and no missing data removal.
  + The poor performance on the correlation=0.8 subset may have been due to the low number of features used for training and testing (=11).

A graph of different colored bars

AI-generated content may be incorrect.

* Interestingly, the XGBoost model had the worst performance when no feature selection was applied, with its best performance on the correlation=0.6 feature subset.

Potentially, the amount of missing data meant that most features were useful.

**Feature Selection Informed by Importance Models Have Placed on Features**:

I used the Random Forest, LightGBM, and XGBoost models that had already been trained on the entire dataset to help select the features with the highest predictive power.

I performed this analysis using the Random Forest, LightGBM, and XGBoost models that had been trained on the full dataset (no missing data removal applied) for each of the cross-validation folds in the two train/test splits (by country and by year). I did not use the models trained with missing data removal because their folds did not have the same columns, so it was not a full comparison. (However, I do have the importance values for the models trained on each dataset).

I found the best features using the permutation importance method. During this method, the values of a single feature were randomly shuffled, and the model was retrained on the dataset. The degradation to the model’s performance on the validation set due to this re-shuffling was noted and used to determine a feature’s importance. As described by Scikit Learn, ‘By breaking the relationship between the feature and the target, we determine how much the model relies on such particular feature.’ This shuffling was performed 10 times per feature. The (score = baseline score minus new score), with the baseline calculated using the full feature set.

One of the major benefits of this method is that it is model agnostic, allowing the same method to be applied to each model type.

After obtaining the feature importance scores for each model/test\_train set/fold combination, I performed the following method:

1. Average and standard deviation of the feature importance scores for the same model across all cross-validation folds.
2. Filter for features with average importance scores minus standard deviation > 0 and average importance scores > 0.001.

*By Country*:

After these processing steps, the LightGBM and Random Forest models were associated with 34 and 32 features, respectively.

After applying these processing steps, the XGBoost model only had 3 features left, and was thus excluded from further analysis (features shown below).

A screenshot of a computer

AI-generated content may be incorrect.

The following features had importance scores meeting the criteria above for both LightGBM and Random Forest:

A white text with black text

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

A table of numbers with text

AI-generated content may be incorrect.

These features were only associated with the LightGBM model after post-processing:

A white text with black text

AI-generated content may be incorrect.

Full LightGBM list of features meeting the processing criteria:

A screenshot of a table

AI-generated content may be incorrect.

These features were only associated with the Random Forest model after post-processing:

A white text with black text

AI-generated content may be incorrect.

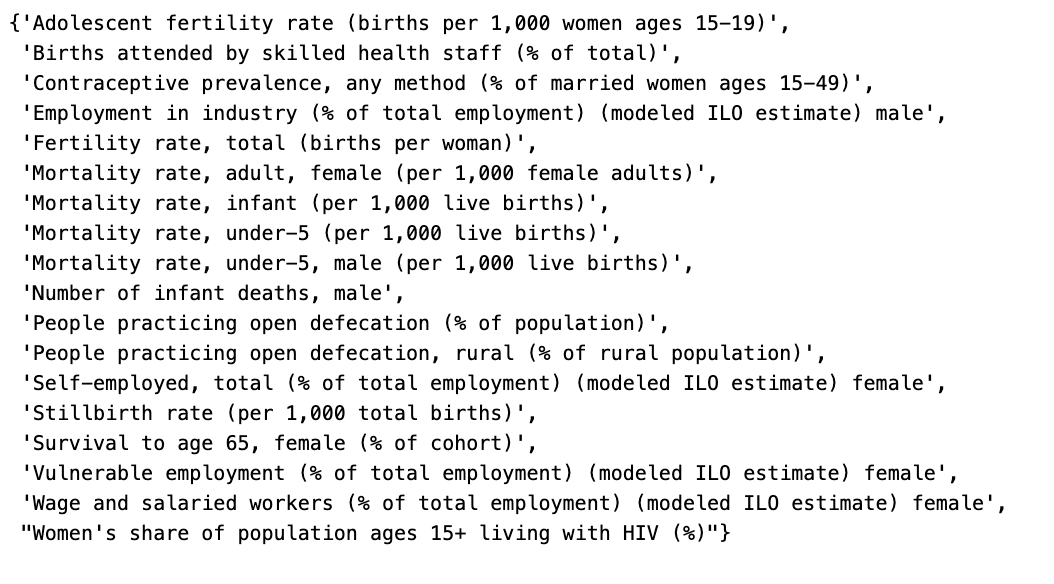
Full Random Forest list of features meeting the processing criteria:

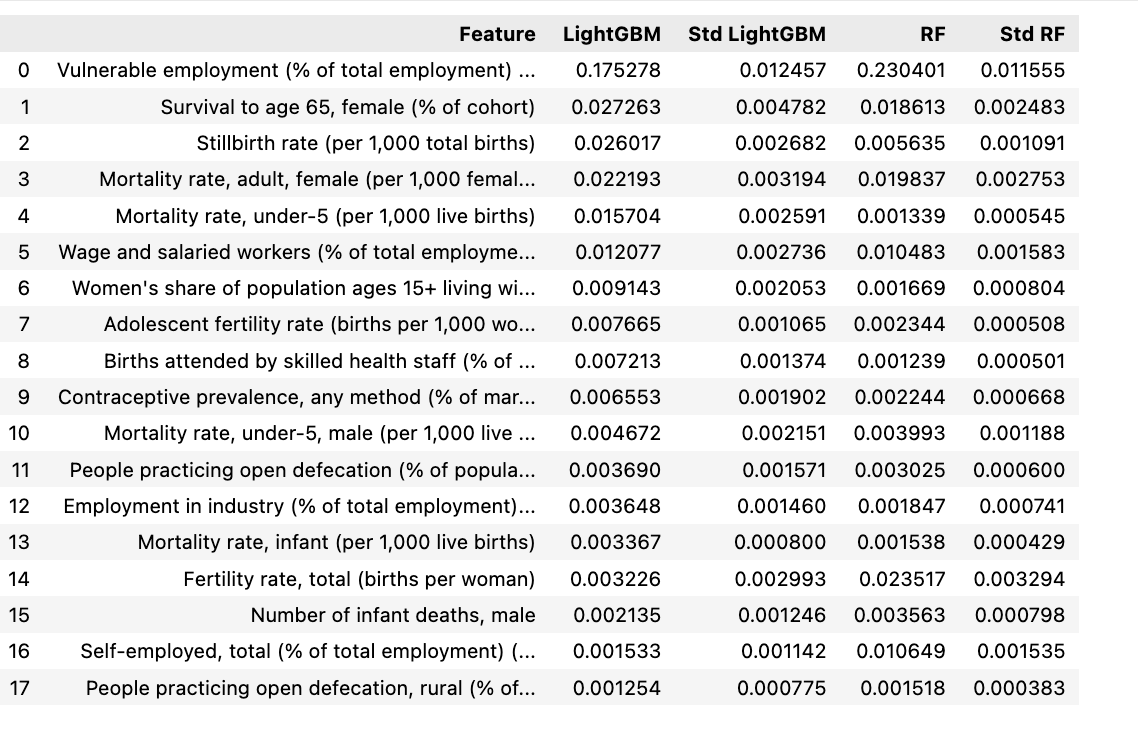
A screenshot of a table

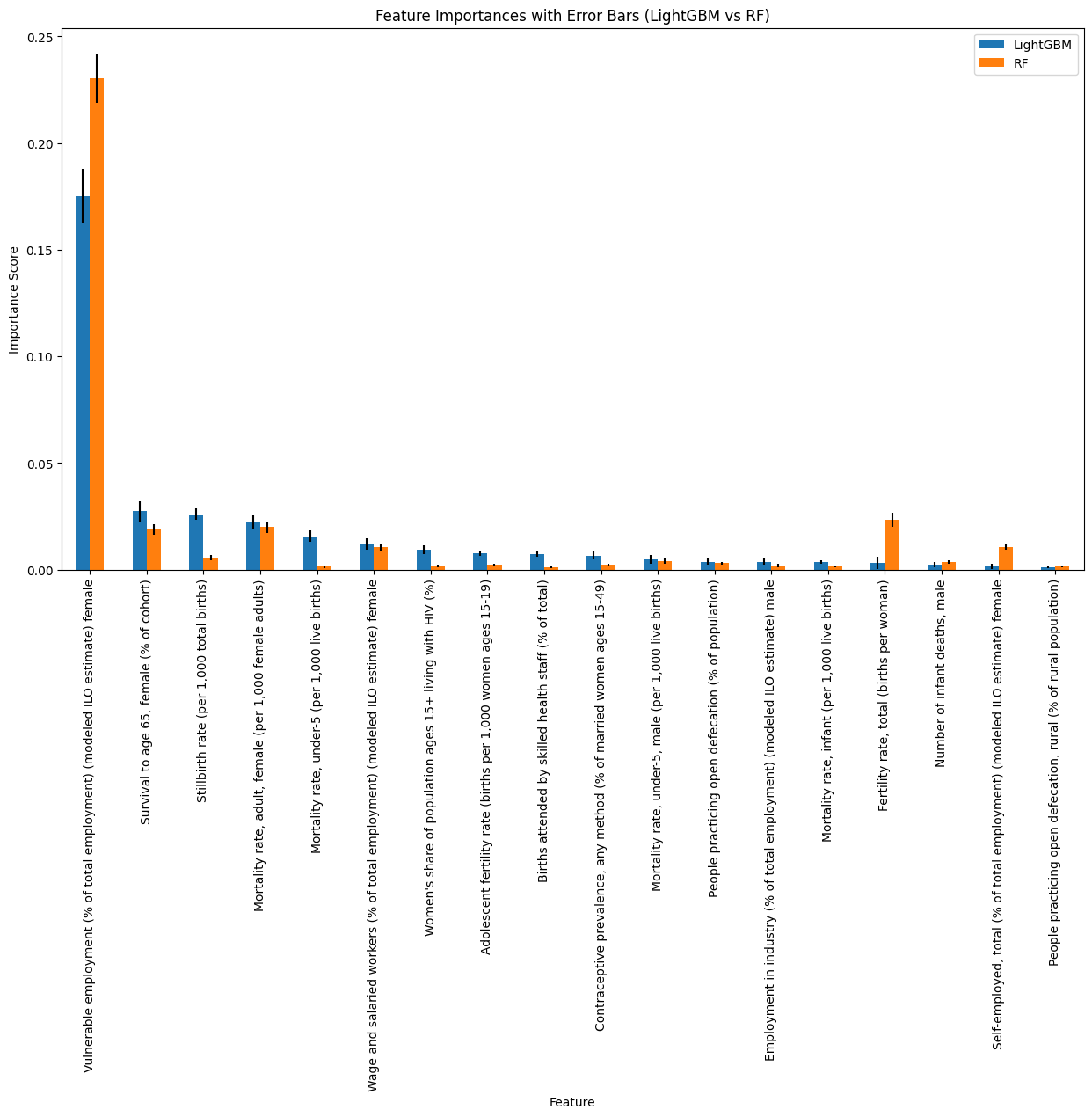
AI-generated content may be incorrect.

*By Year:*

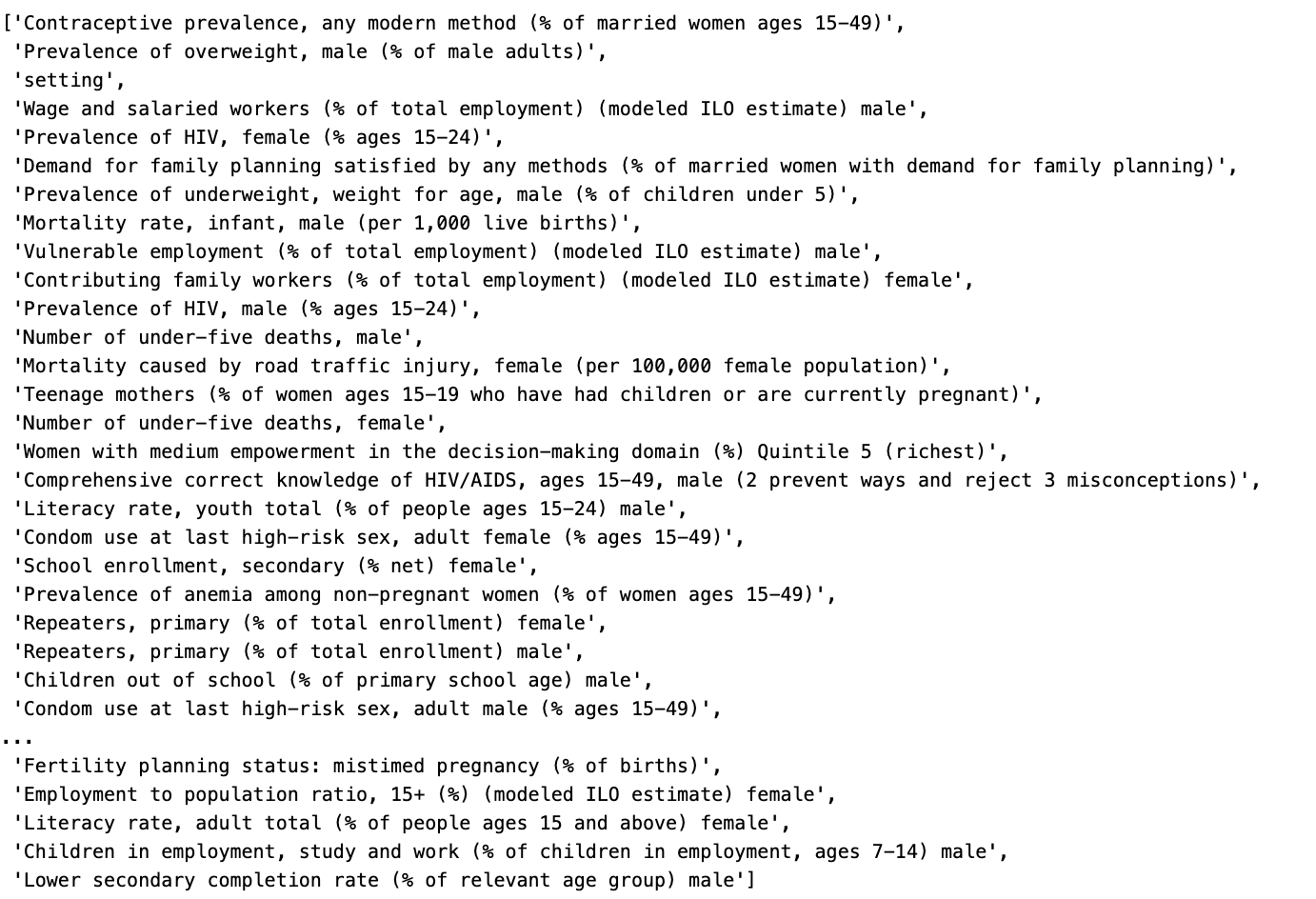
After these processing steps, the LightGBM and Random Forest models were associated with 55 and 27 features, respectively. The intersection between these sets consisted of 18 features.







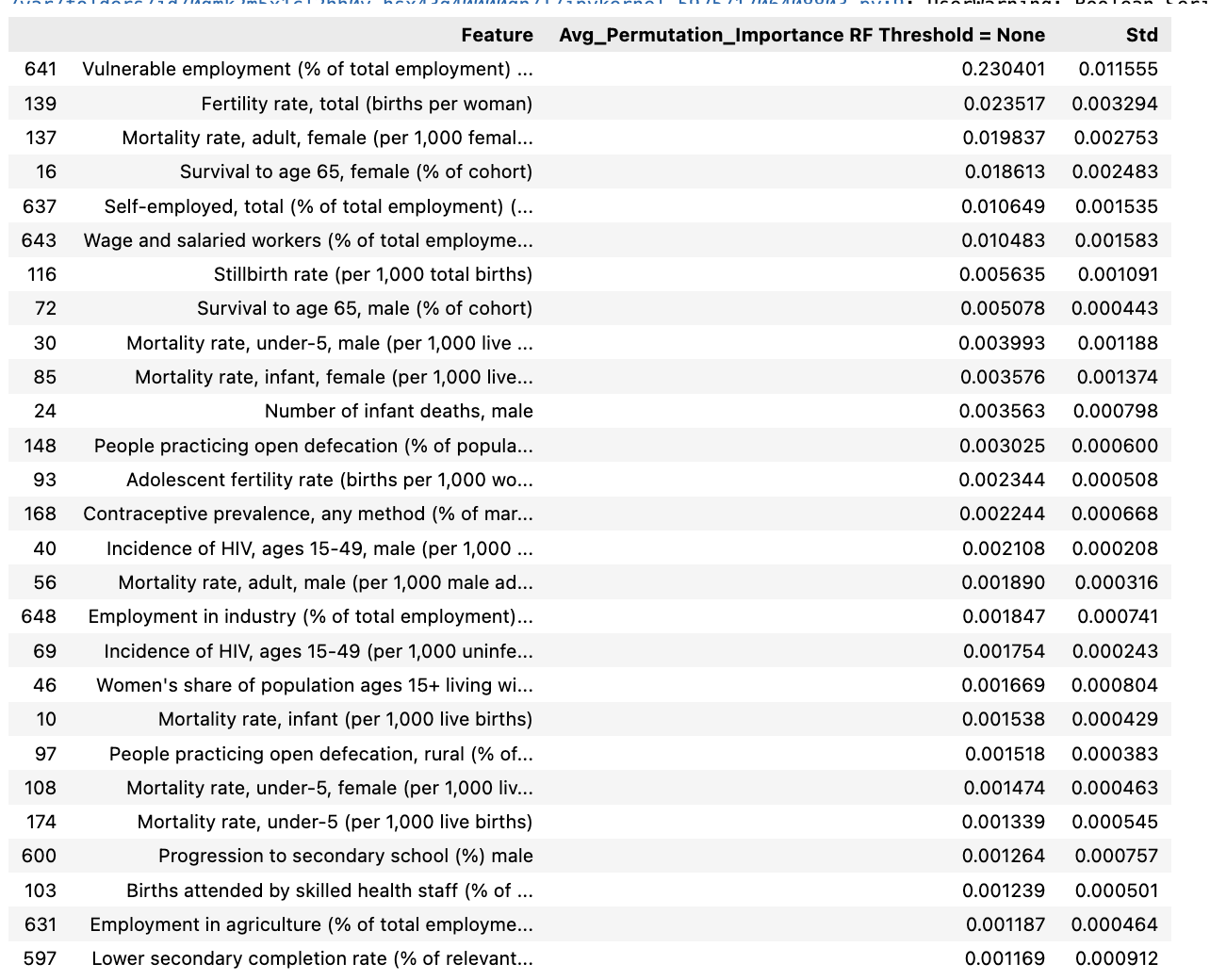
Only LightGBM:





One Random Forest:

A black text on a white background

AI-generated content may be incorrect.

**Potential To-Dos:**

* Implement the model-based feature subsetting method
* Use ensemble methods to further test each subsetting method.