Model Creation and Validation for the Social Vulnerability Index
Training Random Forest and Geographically Weighted Random Forest Models

Thesis for a Master of Public Health, Epidemiology

Nathan Garcia-Diaz

Brown University, School of Public Health

August 26, 2024

Contents

Note: the table of contents acts as in-document hyperlinks

Statement of Purpose

The purpose of the file is to build two final models: a traditional random forest model (RF) and a geographically weighted random forest model (GWFRF). The following two sentence provide a overarching description of the two models. In a RF model, each tree in the forest is built from a different bootstrap sample of the training data, and at each node, a random subset of predictors (features) is considered for splitting, rather than the full set of predictors. A GWRF model expands on this concept by incorporating spatial information by weighting the training samples based on their geographic proximity to the prediction location. The splitting process in a RF model is determined by the mean squared error and in a GWRF is influenced by the spatial weights (i.e., weighted mean squared error), which adjust the contribution of each sample based on its geographic distance.

Overview of Hyperparameters Definitions

In James et al 2021, Ch 8.2.2 Random Forests, James et al 2023, Ch 15.2 Definition of Random Forests and Garson 2021, Ch 5 Random Forest, the hyperparameters that are shared between the traditional RF and the geographically-weighted RF models include:

- **Number of randomly selected predictors**: This is the number of predictors (p) considered for splitting at each node. It controls the diversity among the trees. A smaller m leads to greater diversity, while a larger m can make the trees more similar to each other.
  - for regression this defaults to $p/3$, where p is the total of predictor variables
- **Number of trees**: This is the total number of decision trees in the forest (m). More trees generally lead to a more stable and accurate model, but at the cost of increased computational resources and time.
  - for the `randomForest::randomForest()`, this defaults to 500

Additionally, GWRF involves an extra tuning spatial parameters:

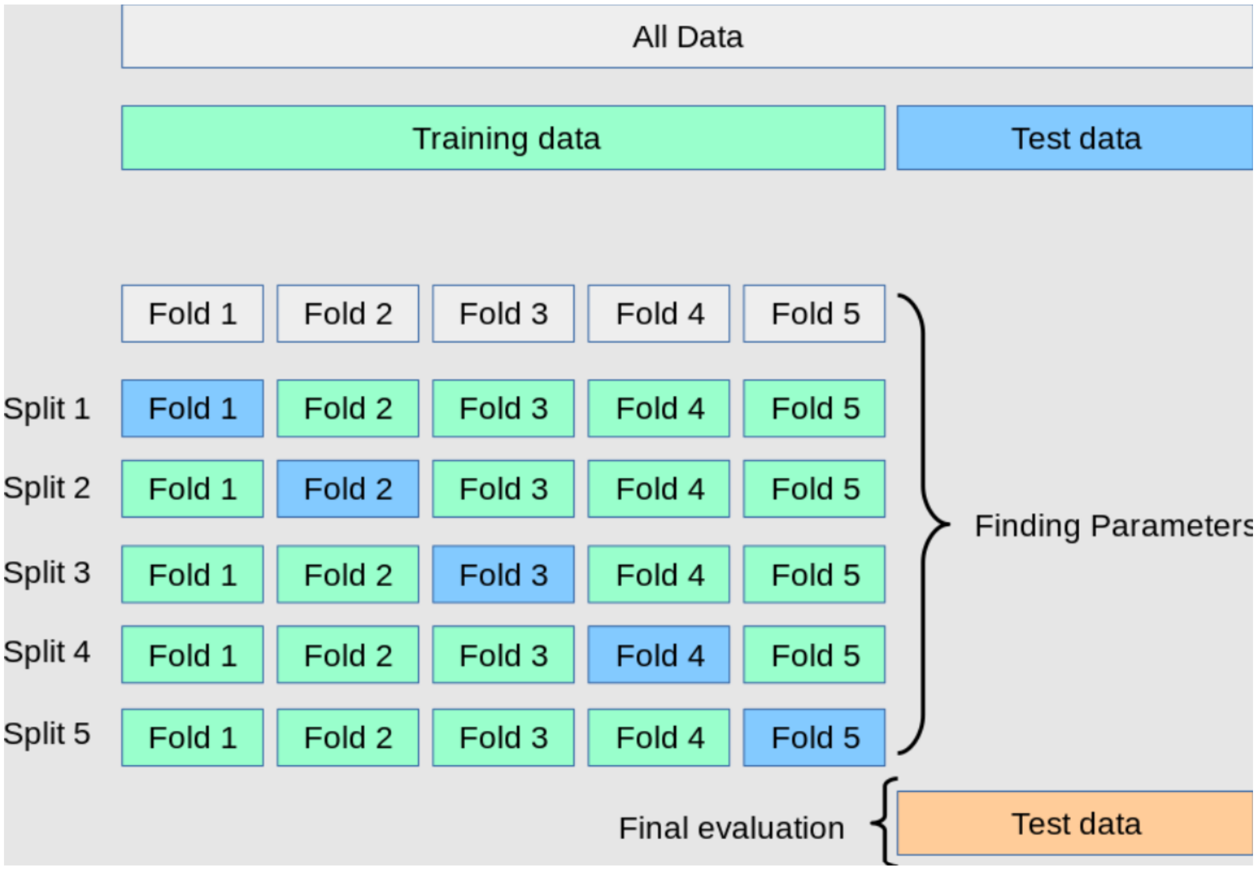
- **Bandwidth parameter**: This controls the influence of spatial weights, determining how quickly the weight decreases with distance. A smaller bandwidth means only very close samples have significant influence, while a larger bandwidth allows more distant samples to also contribute to the model.

Outline of Hyper-parameter Tuning Process

5 RF models will be built, and two metrics will be implemented in the tuning process: Root Mean Squared Error and Out of Bag Error Rate. The models differ based on the different hyperparameters.

(1) default settings
(2) first tune p, and subsequently tune, then m while keeping p constant
(3) simultaneously tune m and p with a grid search
(4) tuned with Out of Bag MSE Error Rates as described by Garson 2021
(5) preform a semi spatial nest cross validation

All models will be validated and tuned with a nested cross-validation, a technique used to assess the performance of a model and tuning hyperparameters. It helps to avoid over fitting and provides an unbiased estimate of model performance. A spatial nested cross-validation is a two-level cross-validation procedure designed to evaluate a model's performance and tune its hyperparameters simultaneously. A nested cross-validation is a method that revolves around an outer and liner loop.A visual description of the method, which can be in Jian et al (2022) - Rapid Analysis of Cylindrical Bypass Flow Field Based on Deep Learning Model



- Outer Cross-Validation Loop:
  - Purpose: To estimate the model's performance on unseen data and provide a more reliable measure of how well the model generalizes to new data.
  - Procedure: The data set is divided into several folds (e.g., 5 or 10). In each iteration, one fold is used as the test set, and the remaining folds are used for training and hyper parameter tuning. This process is repeated for each fold, ensuring that every data point is used for testing exactly once.
- Inner Cross-Validation Loop:
  - Purpose: To select the best hyperparameters for the model.
  - Procedure: Within each training set from the outer loop, a further cross-validation is performed. This involves splitting the training data into additional folds (e.g., 3 or 5). The model is trained with various hyper parameter combinations on these inner folds, and the performance is evaluated to choose the optimal set of hyperparameters.
    Example Workflow:
- Split the data into outer_k folds.
- For each fold in the outer loop:
- Use outer_k - 1 folds for training.
- Apply the inner cross-validation on this training set to tune hyperparameters.
- Evaluate the performance of the model with the selected hyperparameters on the held-out test fold.

- Average the performance metrics across all outer folds to get an overall estimate.

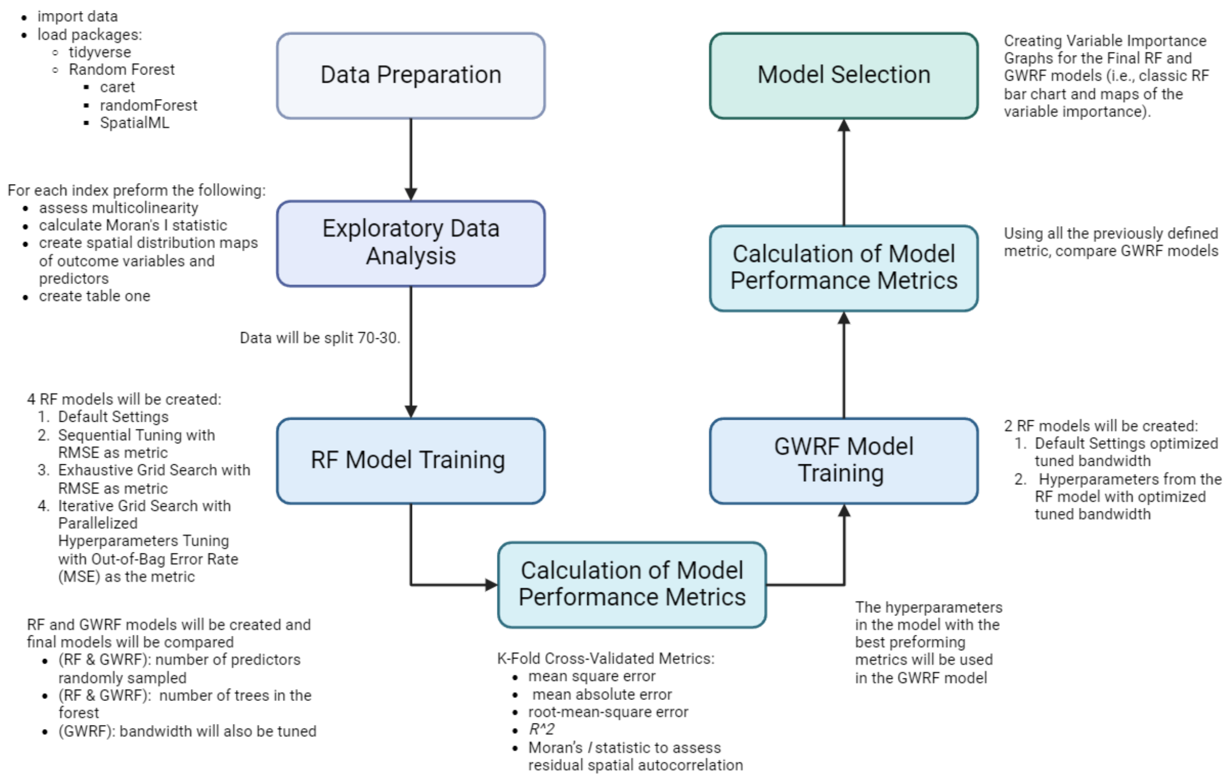
In Garson 2021, Ch 5 Random Forest, Garson teaches Random Forest Models by using `randomForest::randomForest()`, and in chapter 5.5.9 (pg. 267), he provides methods for tuning both of these parameters simultaneously using the Out of Bag MSE Error Rates. This value is a measure of the prediction error for data points that were not used in training each tree, and it can be written as OOB Error Rate $= \frac{1}{n}\Sigma_{i=1}^{N}(y_i - \hat{y}_i^{\text{OOB}})^2$. $\hat{y}_i^{\text{OOB}}$ is the OOB prediction for the i-th observation, which is obtained by averaging the predictions from only those trees that did not include i in their bootstrap sample. To provide a high-level summary, since each tree in a Random Forest is trained on a bootstrap sample (a random sample with replacement) of the data, approximately one-third of the data is not used for training each tree. This subset of data is referred to as the "out-of-bag" data for that tree, and this value is calculated using the data points that were not included in the bootstrap sample used to build each tree.

Georganos et al (2019) created the `package(SpatialML)`, and subsequently the tuning is made possible by the `SpatialML::grf.bw()` function. The function uses an exhaustive approach (i.e., it tests sequential nearest neighbor bandwidths within a range and with a user defined step, and returns a list of goodness of fit statistics).

Summary of Model Building

4 RF models will be built, and they differ based on the different hyperparameters: (1) default settings; (2) tuned by first tuning *mtry*, with *ntrees* set to default, and subsequently tuning then *ntrees* while keeping the newly defined *mtry* constant and both methods use RMSE as the metric; (3) tuned both *mtry* and *ntrees* with an Exhaustive Grid Search and both methods use RMSE as the metric, (4) tune tuned with Out of Bag MSE Error Rates as described by Garson 2021. For each model, MAE, MSE, RMSE, and R^2 will be calculated and the hyperparameters of the best model will continue onto the GWRF. To provide points of comparison in the GWRF, two additional models will be created. Thus, two GWRF models will be created: (1) default *mtry* and *ntrees* with optimized *bandwidth parameter*, and (2) using the previously defined best hyperparameters. The same model evaluation metrics will be compared in addition to calculating the residual autocorrelation.

Lastly, the feature importance plots will be generated for the final, and local feature importance plots will also be created.


• import data
• load packages:
◦ tidyverse
◦ Random Forest
▪ caret
▪ randomForest
▪ SpatialML
Data Preparation
For each index preform the following:
• assess multicolinearity
• calculate Moran's I statistic
• create spatial distribution maps of outcome variables and predictors
• create table one
Exploratory Data Analysis
Data will be split 70-30.
4 RF models will be created:
1. Default Settings
2. Sequential Tuning with RMSE as metric
3. Exhaustive Grid Search with RMSE as metric
4. Iterative Grid Search with Parallelized Hyperparameters Tuning with Out-of-Bag Error Rate (MSE) as the metric
RF Model Training
RF and GWRF models will be created and final models will be compared
• (RF & GWRF): number of predictors randomly sampled
• (RF & GWRF): number of trees in the forest
• (GWRF): bandwidth will also be tuned
Calculation of Model Performance Metrics
K-Fold Cross-Validated Metrics:
• mean square error
• mean absolute error
• root-mean-square error
• R^2
• Moran's I statistic to assess residual spatial autocorrelation
The hyperparameters in the model with the best preforming metrics will be used in the GWRF model
GWRF Model Training
2 RF models will be created:
1. Default Settings optimized tuned bandwidth
2. Hyperparameters from the RF model with optimized tuned bandwidth
Calculation of Model Performance Metrics
Using all the previously defined metric, compare GWRF models
Model Selection
Creating Variable Importance Graphs for the Final RF and GWRF models (i.e., classic RF bar chart and maps of the variable importance).