The effect of random and spatial cross-validation strategies, with spatially lagged variables on extrapolation performance of a random forest model.

Mark Morrison

Acknowledgements

I would like to thank my dissertation supervisor Professor Dirk Husmeier for his invaluable guidance and support throughout this project

Abstract

In many datasets spatial dependence exists such that the values of a random variable for observations are correlated across some locationally referenced space, and this is referred to as spatial autocorrelation (Griffith, 2009). Spatial autocorrelation violates the assumption of independence between observations that is inherent in many procedures used within statistics and machine learning, and one such procedure is the resampling method *k*-fold cross-validation which is commonly used to evaluate predictive modeling performance (Mahoney et al., 2023). Using *k*-fold cross-validation with spatially autocorrelated data can lead to optimistically biased prediction results and poor out of sample area extrapolation performance (Meyer and Pebesma, 2022; Roberts et al., 2017). Spatial cross-validation techniques have been developed to counteract this problem, in which the training and validation data are spatially separated from each other, however random cross-validation is still commonly used in studies when spatial autocorrelation is present (Mahoney et al., 2023).

Methods have also been developed to incorporate spatial dependence within predictive models with the aim of improving prediction performance and reducing the spatial autocorrelation in prediction residuals (Anselin, 2021; Dormann et al., 2007; Nikparvar and Thill, 2021). An approach that has been used with machine learning models has been to include spatially lagged variables in the design matrix used in the model. However, the inclusion of lagged variables in a model introduces the possibility of data leakage during the training process which could lead to overfitting and poor extrapolation performance.

This study aims to investigate the extrapolation performance of a machine learning model using spatial and random cross-validation approaches, with and without spatially lagged variables added. The well-known California housing dataset was used (Universidade do Porto, 2024) as it is known to contain a high degree of spatial autocorrelation in the response variable. A random forest model was used with three cross-validation approaches, random *k*-fold cross-validation, spatial *k*-fold cross-validation with folds created from spatial clusters, and spatial *k*-fold cross-validation with folds created from spatial clusters and additional buffer zones to ensure isolation of training and validation sets. Both spatially lagged response and predictors were included in the models, created using spatial weights matrices based on two distance bands. The three cross validation strategies using combinations of either no lagged variables, lagged response, or lagged predictors and created from the two distance bands resulted in 14 models being evaluated.

To evaluate the extrapolation performance of the models a nested cross-validation approach was used in which the entire dataset was partitioned into five spatially independent clusters with each cluster being used as a test set in a leave-one-group-out cross-validation approach. The outer training sets were then used to train and validate each of the modeling approaches. The difference in mean prediction error between validation and testing was then used to identify evidence of overfitting and potential data leakage.

The only significant difference between validation and test error was found in three of the random cross-validation models with lagged variables included, with each showing optimistically biased validation results. Furthermore, for each of these three models, a lagged variable had the highest variable importance score. These two points provide preliminary evidence in support of data leakage occurring when lagged variables are used with random cross- validation in the presence of spatial autocorrelation in the response variable.

Contents

1. Introduction

In many fields such as ecology, econometrics, and geography (Cliff and Ord, 1970; Elhorst, 2014; Legendre and Fortin, 1989) spatial dependence often exists within georeferenced data where the values for a variable at a given location are related to those at neighbouring locations. The phenomenon where a variable is correlated with itself over space, is commonly referred to as spatial autocorrelation (Griffith, 2009). The presence of spatial autocorrelation within a dataset has implications for the predictive modeling process. Many statistical and machine learning processes assume data is independent and identically distributed (i.i.d), and when spatial autocorrelation is present this assumption can be violated (Anselin, 2002; Mahoney et al., 2023).

An example of a commonly used resampling technique used to evaluate predictive model performance that relies on the i.i.d assumption is *k*-fold cross-validation (Kuhn and Johnson, 2013). *k*-fold cross-validation involves randomly splitting the data into *k* equally (approximately) sized folds and using each fold as a validation set on which the predictive performance of a model, that has been trained on the remaining folds, is assessed. The random assignment of observations to a fold relies on the assumption that they are independent from one another, and in the presence of spatial autocorrelation, this is not always the case as some observations in the training set will have spatial neighbours in the validation set, to which they are related. This creates a situation where the validation set does not truly represent "unseen" data and model predictive performance is overly optimistic and fails to generalize well to new data that is spatially independent from that used during training (Mahoney et al., 2023, 2023; Schratz et al., 2019). To deal with this problem cross-validation methods have been devised in which the validation data is kept isolated from the training data and collectively these have been referred to as spatial cross-validation strategies (Mahoney et al., 2023). Despite the fact that spatial cross-validation has been consistently shown to produce more realistic out of sample predictions compared with random cross-validation when spatial dependence exists in the data, random cross-validation is still used by the majority of machine learning studies (Meyer and Pebesma, 2022).

Many statistical techniques have been developed that aim to incorporate spatial dependence in the predictive modelling process with the goal of improving the accuracy of predictions (Li, 2022) and to reduce spatial dependence in the model residual errors (Dormann et al., 2007). Historically this has included methods such as autoregressive models, spatial eigenvector mapping, generalized least squares, spatial generalized linear mixed models, spatial generalized estimating equations, geographically weighted regression, and kriging (Anselin, 2021; Dormann et al., 2007; Miller et al., 2007). All these models incorporate spatial dependence in different ways, either through the inclusion of spatially lagged terms (Anselin, 2021), spatial random effects (Melo et al., 2016), spatial covariance structures (Dormann et al., 2007), or through explicit spatial interpolation techniques (Cressie, 1990).

More recently with the increase in big data containing a spatial element there has been growing interest in using machine learning techniques to make predictions with data containing spatial dependence (Sabek and Mokbel, 2020). Machine learning methods are popular due to their accuracy, ability to reveal non-linear relationships, ability to deal with correlated data, and fewer model assumptions (Cai et al., 2023; Schratz et al., 2019). As with traditional statistical methods many techniques have been implemented that aim to account for spatial dependance within machine learning models. Nikparvar and Thill (2021) identified that generally the inclusion of spatial dependance within machine learning frameworks was either through its representation in the observation design matrix or incorporation into the learning algorithm. Examples of the latter include predictive clustering trees with a spatial autocorrelation included in the node splitting test (Stojanova et al., 2011), geographically weighted random forests (Quiñones et al., 2021), support vector random fields that explicitly model spatial dependence using a conditional random field framework (Lee et al., 2005).

The inclusion of spatial dependence within the observation design matrix is often a simpler approach and allows for the use of standard machine learning techniques without the need for additional modification of the algorithm. One of the most common approaches in this regard has been to include spatial features such as map coordinates (Walsh et al., 2017; Wang et al., 2017), distance to points within the sample space (Hengl et al., 2018), or distance to boundaries of the model area (Behrens et al., 2018). While inclusion of features such as spatial coordinates can improve model predictive performance Meyer and colleagues (2019) found that they led to model overfitting when predicting outside the model training space.

Another method that has been used to incorporate spatial dependence within the observation design matrix has been the addition of spatially lagged variables, with either the spatial lag of the response (Liu et al., 2022; Pawley et al., 2024; Schneider et al., 2020), the predictors (Hengl et al., 2018; Kämäräinen et al., 2023; Kiely and Bastian, 2020), or both (Credit, 2022). Generally, this has led to improvements in predictive performance over models without lagged variables included (Credit, 2022; Kiely and Bastian, 2020; Liu et al., 2022; Pawley et al., 2024), however a potential issue exists when lagged variables are added to a model in the form of data leakage from the validation set into the training set during model fitting. Data leakage occurs when information from the test or validation data sets is unintentionally included in the model training process at some point during data collection, sampling, or pre-processing and has been identified as a serious problem in many machine learning studies as it leads to overly optimistic results (Kapoor and Narayanan, 2023). Spatially lagged variables include information about neighbouring locations and when observations from the training set have neighbours in the validation set data leakage can occur. To the best knowledge of the author the issue has yet to be addressed by studies that have included spatially lagged variables in the modelling process and is a question that requires further investigation.

2. Literature review

2.1. Measuring spatial autocorrelation

The degree to which spatial autocorrelation is present in a dataset is commonly represented by the global Moran's I statistic (Moran, 1948) or a variogram (Matheron, 1963). The Moran's I is a global measure of spatial autocorrelation that quantifies the similarity of values for a variable at different locations and defined as

$$I = \frac{n}{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}} \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(y_i - \bar{y})(y_j - \bar{y})}{\sum_{i=1}^{n}(y_i - \bar{y})}$$

Where the $n \times 1$ vector of observations $\mathbf{y}$ is the variable of interest, $\bar{y}$ is the mean of $\mathbf{y}$, w_{ij} represents the value of the i-th row and j-th column of an $n \times n$ spatial weights matrix $\mathbf{W}$ and n represents the number of observations. Values for Moran's I range from -1 to 1 with positive values of indicating spatial clustering of similar values for a variable in space and negative values clustering of dissimilar values of a variable. The spatial weights matrix $\mathbf{W}$ describes the spatial relationships between observations and the method used to define these relationships can impact the value for I. Common choices for the defining $\mathbf{W}$ are; contiguity-based matrices, where observations are neighbours if they share a common boundary, distance-based matrices which designate observations within a distance threshold as neighbours, inverse distance weights which are inversely proportional to the distance between observations, k-nearest neighbours weights which consider an observation to have exactly k neighbours defined as the k closest observations, and finally block weights matrices which are based on an observations membership to some pre-defined block such as a region or county (Rey et al., 2023). Typically the weights matrix is row normalised so that the each row sums to one and the weighting process effectively produces an average of neighbouring values (Elhorst, 2014).

The semivariogram, typically referred to as a variogram, is a tool from the field of geostatistics that models the spatial correlation in a dataset by plotting the semivariance of a variable as a function of distance (Bivand et al., 2008). The variogram typically assumes second-order stationarity, implying that the mean and variance are constant within the spatial field, and the covariance between points is dependent solely on their separation distance h, not their specific locations (Webster and Oliver, 2007). The variogram is defined as

$$\gamma(h) = \frac{1}{2}\mathrm{E}[(Z(s) - Z(s+h))^2]$$

Where $Z(s)$ is a function that generates the observation at location s with constant mean and variance. A variogram calculated from sampled data is referred to as an empirical variogram and in practice it is common to use binned distances $\tilde{h}_j$. The variogram is defined as

$$\tilde{\gamma}(\tilde{h}_j) = \frac{1}{2N_h}\sum_{i=1}^{N_h}(Z(s_i) - Z(s_i + h))^2, \qquad \forall h \in \tilde{h}_j$$

Where N is the number of sample data pairs. The degree of similarity over distance is expressed by the covariance function $C_z(\tilde{h}_j)$ which is defined as

$$C_z(\tilde{h}_j) = \sigma_z^2 - \tilde{\gamma}(\tilde{h}_j)$$

Where σ_z^2 represents the stationary variance. The point at which $C_z(\tilde{h}_j)$ is equal to zero is called the sill and the lag distance at which the variogram reaches the 95% of the sill is called the range (Salazar et al., 2022). The range and sill are typically obtained after fitting a variogram model to the empirical variogram, with typical models including the Spherical model, the Exponential model, the Gaussian model and the Matern model (Webster and Oliver, 2007).

2.2. Spatial cross-validation methods

Several methods have been developed to isolate data used to train models from those used to validate them based on their spatial location. These include; dividing the spatial area into blocks based on a grid placed over the sample area, with all observations within a block being assigned to a fold (Roberts et al., 2017; Wenger and Olden, 2012), creating spatial clusters using K-means which are then used as folds (Brenning, 2012; Schratz et al., 2019) and including an exclusion buffer zone around individual (Pohjankukka et al., 2017) or grouped validation observations

(Wang et al., 2023). A visual representation of spatial-cross validation based on groups formed using K-means clustering with the addition of a buffer zone is displayed in Figure 1 . In a study using simulated data Mahoney and colleagues (2023) compared the predictive performance of a number of spatial cross-validation methods and found that spatially clustered folds, and the use of exclusion buffers around validation sets produced the most accurate validation results. Using spatial clusters rather than blocks has the advantage of producing more sensible fold boundaries with fewer observation along the fold edge, and the use of exclusion buffers around folds can ensure validation and training data are separated by the range over which spatial dependency extends (Mahoney et al., 2023). Creating cross-validation folds based on spatial clusters should also offer a solution to the problem of data leakage from lagged predictors as fewer observations from the validation set would have neighbours in the training set. Adding a buffer zone around the validation set that extends beyond the distance used to define neighbour relationships should then remove this problem entirely.

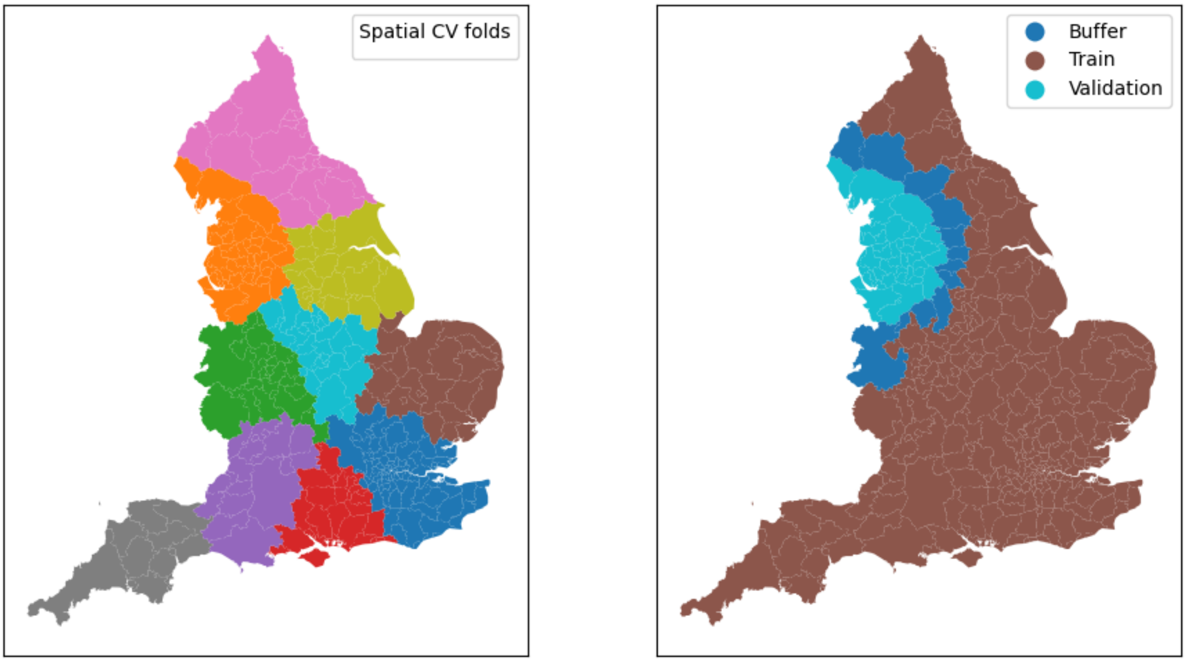


Figure 1. Visual representation of a spatial cross-validation approach utilising K-means clustering and the addition of a buffer zone. Left plot. The observations (English Local Authorities) are split into folds using K-means. Right plot: Example of a fold being used as validation set with buffer zone applied.

2.3. Spatially lagged variables in machine learning models

Lagged variables are generally created using a spatial weights matrix like those used to calculate Moran's I such that for the variable $\mathbf{z}$ the spatial lag $\mathbf{z}_l$ is defined as

$$\mathbf{z}_l = \mathbf{W}\mathbf{z}$$

With $\mathbf{W}$ being the $n \times n$ spatial weights matrix and $\mathbf{z}$ an $n \times 1$ vector of observations (Elhorst, 2014).

Several studies have compared the performance of machine learning models, with and without lagged variables added. Kiely and Bastin (2020) included spatially lagged predictors in random forest, gradient boosting machine and neural network models in a study predicting New York real estate prices. They found that including spatial lags of predictors calculated using 500m distance band improved prediction performance in two of the three models (gradient boosting machine and neural networks) compared to models without lagged predictors added. A study using machine learning approaches to map depth to bedrock found that including spatial lags of the response variable calculated from distance weighted mean values of the 5, 10 and 15 nearest neighbour's significantly improved performance over models with only auxiliary predictors and geographic coordinates (Pawley et al., 2024). In predicting employment around a transit line in Los Angeles, Credit (2022) found a marginal improvement in prediction accuracy when lagged response and predictor variables were added to a random forest model compared to the model with unlagged predictors. Lagged variables were created from average values from the 12 nearest neighbours. Liu et al. (2022) evaluated the performance of random forest models with and without lagged response variables added using two well know publicly available machine learning datasets (Meuse and California housing). They included spatial lags of the response variable averaged over 5, 10 and 15 nearest neighbours for the Meuse data and 5, 10, 15, and 50 nearest neighbours for the California housing data. The authors reported lower prediction error when the lagged response was included in the models as well as up to a 95% reduction in spatial autocorrelation of residuals.

Other studies have utilised lagged variables in machine learning models without including a direct comparison with models without lagged variables included. In a study aimed at using machine learning to reconstruct daily $PM_{2.5}$ concentrations across Great Britain, Schneider et al. (2020) included spatial lags of the response variable, created using inverse distance weights, to a random forest model along with other predictors. Kämäräinen et al. (2023) included spatially and temporally lagged predictors to random forest and gradient boosting models in the prediction of atmosphere-forest CO_2 exchange, using the 25 nearest locations to create spatially lagged predictors. Hu et al. (2017) also include a lag of the response variable to model $PM_{2.5}$ concentrations using a random forest model, they did not report how the lagged variable was determined other than stating nearby $PM_{2.5}$ concentrations were included in the predictors set.

Although the results of the studies suggest that inclusion of spatially lagged variables in machine learning models can improve predictive performance, concerns exist about the generalisation of performance to locations outside of the study area. None of the aforementioned studies (with the exception of Hu et al. (2017)), utilized spatial cross-validation in the evaluation of model predictive performance, and no studies assessed the extrapolation performance of the models into test locations independent of the training/validation area. Furthermore, none of the studies have accounted for potential data leakage caused by the inclusion of spatial lagged predictions during model fitting. This is of particular concern in the studies by Hu et al. (2017) and Liu et al. (2022) that used random forest models, with both reporting lagged response variables as having the highest variable importance scores out of the set of predictors.

2.4. Project aims

This study aims to investigate the out-of-sample-area extrapolation performance of a machine learning model using spatial and random cross-validation approaches, with and without spatially lagged variables added. And secondly, to investigate the effect of adding buffer zones around validation test sets to mitigate potential data leakage from lagged predictors during model training.

3. Data

The data used for this project was the publicly available and widely used California housing data set first published by Pace and Barry (1997) and was obtained from the Universidade do Porto website (2024). It was chosen as it is known to contain strong spatial autocorrelation in the response variable, and has been used by other studies examining techniques for mitigating spatial autocorrelation (Liu et al., 2022; Wang et al., 2023) which allows for some methodological comparisons. The dataset contains 20640 observations with information regarding census block groups from the 1990 U.S. Census for the state of California and typically median house price is used as the response variable. The original variables are described in Table 1, there are no missing values, and all variables are numeric.

Table 1. Description of original variables from the California housing dataset

Variable	Description	Data type
longitude	Geographic coordinate indicating east-west position of block centroid	numeric
latitude	Geographic coordinate indicating north-south position of block centroid	numeric
housingMedianAge	Median age of houses within a block	numeric
totalRooms	Total number of rooms within a block	numeric
totalBedrooms	Total number of bedrooms within a block	numeric
population	Total number of people living within a block	numeric
households	Total number of households within a block	numeric
medianIncome	Median household income for a block in tens of thousands	numeric
medianHouseValue	Median house value for a block	numeric

The longitude and latitude were not included as predictors in the models due to the issues outlined by Meyer et al. (2019). Some feature engineering was performed to reduce the high degree of collinearity between predictors (see correlation matrix Appendix 1 Figure 1) such that ‘totalRooms’, ‘totalBedrooms’, and ‘population’ were all divided by the number of households to create average values per household. Six variables ('medianHouseValue', 'medianIncome', 'averageRooms’, ‘averageBedrooms', 'popPerHouse' and 'population') were heavily skewed by outliers (see histograms Appendix 1 Figure 2) and so these variables were log-transformed. A correlation matrix and histograms for the variable set after feature engineering and log-transformation are displayed in Appendix 1 Figure 3 and Appendix 1 Figure 4 respectively. The list of final variables used in the modelling process is displayed in Table 2.

Table 2. Description of variables used for modelling

Variable	Description	Data type
housingMedianAge	Median age of houses within a block	numeric
log_medianHouseValue	Log of median house value for a block	numeric
log_medianIncome	Log of median income for households in a block	numeric
log_averageRooms	Log of average rooms for households per household in a block	numeric
log_averageBedrooms	Log of average bedrooms for households in a block	numeric
log_popPerHouse	Log of population per household for a block	numeric
log_population	Log of population for a block	numeric

The spatial distribution of the response variable is displayed in Figure 2 indicating the clustering of similar values, especially for higher values around the coastal areas and cities.

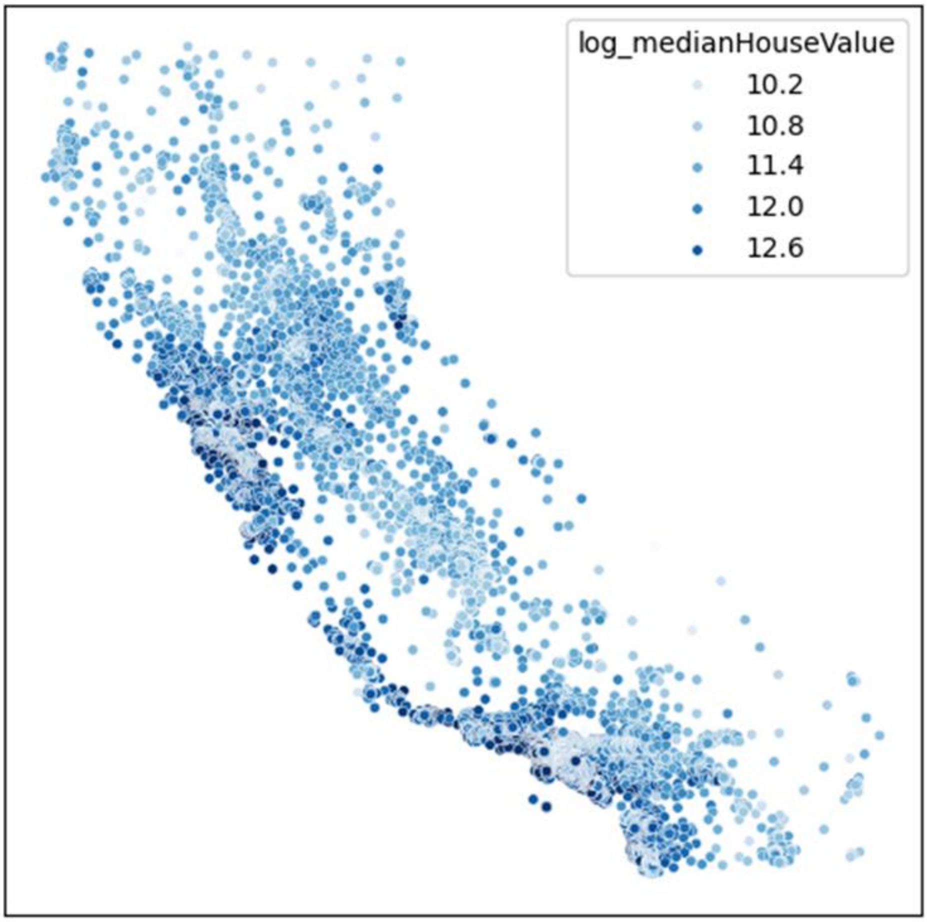


Figure 2. Spatial distribution of the response (log_medianHouseValue) variable indicating clustering of higher values around the coastal areas and cities.

Prior to the creation of lagged variables the coordinate reference system of the data set was changed to the Spherical/Web Mercator (EPSG:3857) projected coordinate system ("Web Mercator projection," 2024) which has a unit of measurement in meters for ease of interpretation of buffer distances during modeling.

4. Method

4.1. Random forest model

A random forest model was chosen as the machine learning approach as it has been effectively used in several studies (Liu et al., 2022; Mahoney et al., 2023; Schratz et al., 2019; Wang et al., 2023) looking at approaches to mitigate spatial dependence in machine learning applications. Random forest models have also been reported to perform well without extensive hyperparameter tuning (Hastie et al., 2009; Probst et al., 2019), and this was an important consideration in this study as the dataset is large and the computing resources limited. The random forest model was first devised by Breiman (2001) and is based on the concept of averaging a number of randomized decision trees, specifically regression tress in the context of this study. A regression tree applies recursive binary splitting to partition the predictor space into regions (Hastie et al., 2009), and the random forest model employed in this study utilised the CART method introduced by Breiman et al. (1984). The CART splitting algorithm works by selecting the predictor X_j from the set of predictors and the splitting point s such that splitting the predictor space into the regions $R_1(j,s)$ and $R_2(j,s)$, defined as

$$R_1(j,s) = \{X|X_j < s\} \text{ and } R_2(j,s) = \{X|X_j \geq s\}$$

minimizes the residual sum of squares RSS given by

$$RSS = \sum_{x_i \in R_1(j,s)} (y_i - \bar{y}_{R_1})^2 + \sum_{x_i \in R_2(j,s)} (y_i - \bar{y}_{R_2})^2$$

where $\bar{y}_{R_1}$ and $\bar{y}_{R_2}$ are the means of the dependent variable for the observations in $R_1(j,s)$ and $R_2(j,s)$ respectively (Hastie et al., 2009). It is worth noting that an alternative to using a piecewise constant model (as described above) for determining splits is a piecewise linear model such as the M5 model (Quinlan, 1992) which uses linear regressions at the terminal leaf nodes to estimate the dependent variable. The piecewise constant approach was used in the random forest models for this study as this was the only option within the modelling package that was used.

The random forest algorithm involves taking the results of many regression trees and averaging them. The keys to the algorithm's success are that each individual tree is built from bootstrapped samples of the data and the predictors selected to make each decision split are taken from a subset of the total set of predictors (James et al., 2021). The latter point reduces the correlation among the trees which in turn reduces the variance of the average. As explained by Hastie et al (2009), the average variance of a collection of i.i.d. random samples is the common variance divided by the number of samples. However, as the collection of trees in a random forest are constructed from bootstrapped samples of the same dataset they are identically, but not independently distributed and as such the average variance is defined as

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

where ρ is the positive pairwise correlation, B the number of trees, σ^2 the variance of the individual trees. With large number of trees reducing the correlation between them has the greatest impact on reducing the average variance and limiting the number of predictors available for selection at each split in each tree in the random forest does just that.

The random forest regression algorithm is defined by Hastie et al. (2009) as follows

1. For $b = 1$ to B
   a. Take a bootstrap sample $\mathbf{Z}$ of size N from the training data
   b. Grow a regression tree T_b using the bootstrapped data by recursively repeating the following steps for each terminal node until the minimum node size is reached.
      i. Randomly select m predictors from the full predictor set

ii. Select the best combination of predictor and split point from the m predictors

iii. Split the node into two child nodes

2. Return the ensemble of trees $\{T_b\}_1^B$

Predictions for the value at new point x are made by applying

$$\hat{f}_{rf}^{B}(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$$

where B is the number of trees in the forest and $T_b(x)$ is the tree at point x.

Many machine learning models have hyperparameters that require tuning as they cannot be determined directly from the data (Kuhn and Johnson, 2013). While the random forest model has several hyperparameters that can be tuned this study limited tuning to the number of predictors to choose from at each split, commonly referred to as m_{try}, as this is considered the main hyperparameter of the model for the reasons explained above (Kuhn and Johnson, 2013). Tuning additional hyperparameters would also have increased the computational cost beyond the capacity of the computing resources available for this study. The values of m_{try} ranging from one to six were used for all models in this study. The total number of trees is sometimes tuned, but as random forests do not overfit with additional trees, utilizing a sufficiently large number of trees can avoid the need for tuning this hyperparameter (James et al., 2021). The default setting for the number of trees in the software package used in this study was 500 and this was increased to 1000 for all models. To gain an understanding of how much influence lagged predictors had on the models the mean increase in node purity for each predictor at each occurrence of that predictor within the forest was recorded. Node purity is a measure of how well the nodes separate the data with respect to the dependent variable and in the context of regression trees is the decrease in the RSS. The variable importance is then the amount the RSS decreases as a result of using a given predictor averaged for all splits across all trees (James et al., 2021), and this study was expressed a percentage of the most important variable.

4.2. Lagged variables and the assessment of spatial autocorrelation

Distance bands were chosen as the method to define the neighbour relationships for the spatial weight's matrix $\mathbf{W}$ such that w_{ij} is the element of $\mathbf{W}$ that expresses the neighbour relationship between locations i and j and is defined as

$$w_{ij} = \begin{cases} 1, & if\ d_{ij} \leq \delta \\ 0, & otherwise \end{cases}$$

where d_{ij} is the Euclidean distance between locations i and j and δ is the pre-defined distance band threshold. As per convention, diagonal entries which represent an observation's neighbour relationship with itself, were set to zero and rows of $\mathbf{W}$ standardized so they sum to one, so that $\mathbf{W}$ represents a local average of the neighbouring values (Elhorst, 2014). The distance band specification was chosen to define neighbour relationships for the following reasons. To evaluate the impact of buffer zones added to clusters during the spatial cross-validation resampling approach, the maximum distance of each observation's furthest neighbour needed to not exceed the buffer distance. This excluded the k-nearest neighbours specification because the maximum nearest neighbour distance for observations within this dataset was over 80km which would have required overly large buffer zones. Contiguity weights could not be used as this would require information about the geographic boundaries of the census blocks which was not available. Inverse distance weights were initially going to be used as they account for decline in spatial autocorrelation with distance. However, it was found that many observations had the same point centroid locations due to the resolution of the coordinates being low (two decimal places in latitude and longitude giving an accuracy of approximately 1km) and as such many observations had neighbours that were zero distance away. This was a problem as the software (Rey and Anselin, 2007) used to create the lagged variables calculates inverse distance weights as the separating distance d_{ij} raised to the power of a decay parameter. If inverse distance weights had been used under these circumstances any neighbours for an observation separated by zero distance would not have been included in the creation of lagged variables.

To identify the extent of the spatial autocorrelation in the response variable an empirical variogram was constructed for the response variable using 20 lags up to a maximum distance of 100km and was then fit with four variogram models (Spherical, Exponential, Gaussian and Matern). The root mean square error (RMSE) for the fit, along with visual inspection of the variogram plots was used to obtain the range from the best fitting model (see Figure 3). The Spherical, Exponential, and Matern models had the same RMSE (0.017), but the Spherical model fit the most data points from the empirical variogram and so the range for this model (23934m) was deemed reflective of the extent of the spatial autocorrelation in the response variable.

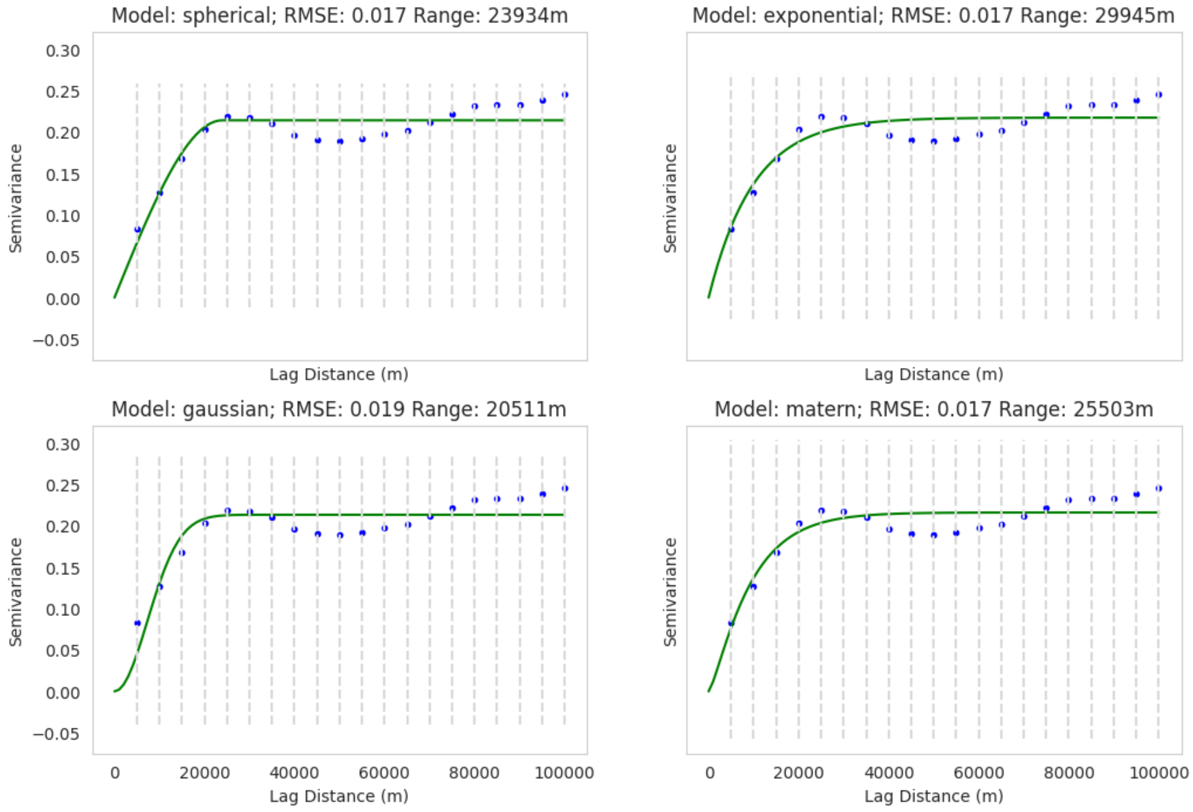


Figure 3. RMSE and range for Spherical, Exponential, Gaussian and Matern models fit to an empirical variogram constructed for the response variable using 20 lags between 0 and 100km.

To determine the size of the distance bands for the construction of lagged variables the Moran's I statistic was calculated for the response variable using distance bands from 2500m to 50000m (see Figure 4 and Table 3). The significance of the Moran's I statistic was evaluated by constructing a simulated p-value obtained by comparing the obtained statistic to a distribution generated from 999 randomly shuffled permutations of the data. For all distance bands the Moran's I statistic was highly significant (p = 0.001). As expected, a high degree of spatial autocorrelation was found for the response variable and was largest when the smallest distance band of 2500m was used (Moran's I = 0.789) and declined with increasing distance. The greatest change in spatial autocorrelation occurs between 2500m and 25000m suggesting that including neighbours beyond 25000 made little change to the spatial autocorrelation, which is consistent with the variogram results. As no information about the spatial autocorrelation data generating process was available it was decided that two distance bands would be used in the modelling process, one smaller where the magnitude of spatial autocorrelation was greatest, and one larger closer to the limit of the spatial autocorrelation identified by the variogram. For the smaller distance band, 5000m was chosen over 2500m as the later had a higher proportion (12% vs 5%) of observations without neighbours (called islands) while the former still had a high degree of spatial autocorrelation at 0.728 (see Table 3). 20000m was chosen for the larger distance band as it sits just within the range of the variogram. The use one small and one larger distance band would also help to shed light on the potential impact of data leakage from lagged variables. The larger distance band included on average 10 times the number of neighbours as the smaller one and so if data leakage was to occur it should be more pronounced with the lagged variables created using the larger distance band. Spatial weights matrices based on these two distance bands were then used to create spatial lags of the predictors and the response variable which were then used in the modelling process.

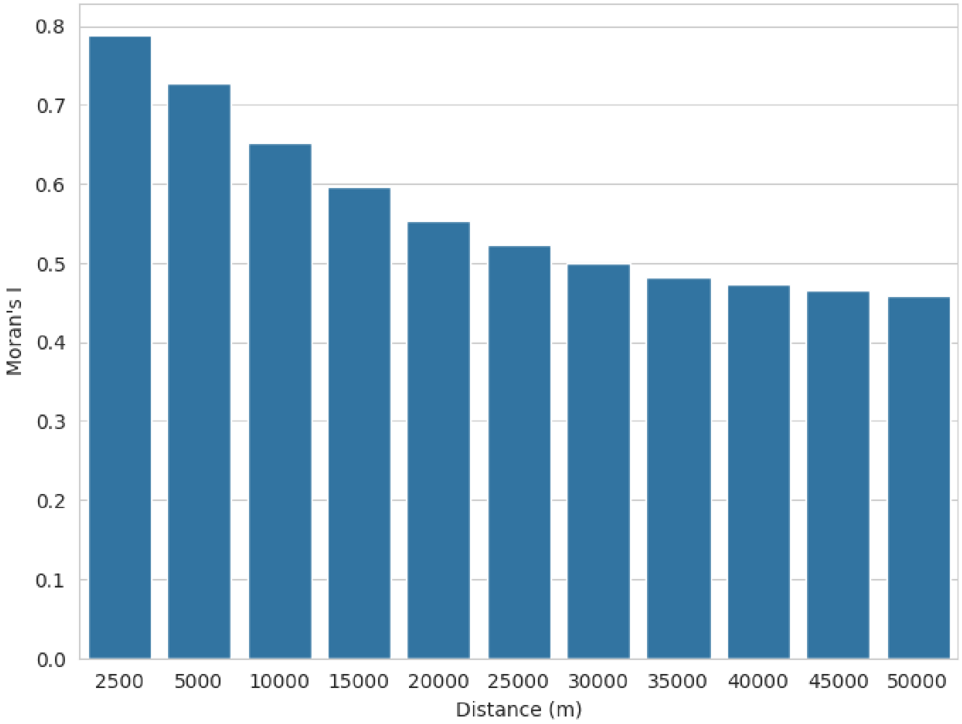


Figure 4. Moran's I for the response variable using distance band weights ranging from 2500m to 50000m

Table 3. Global Moran's I calculated for distance bands between 2500m and 50000m. The simulated p-value for Moran's I is obtained by comparing the observed Moran's I value to a distribution generated from 999 randomly shuffled permutations of the data. Islands are observations without neighbours.

Distance band (m)	Global Moran's I	Simulated p-value	Number of islands	Maximum neighbours	Minimum neighbours	Mean neighbours
2500	0.789	0.001	2507	87	0	16
5000	0.728	0.001	1111	309	0	69
10000	0.653	0.001	457	862	0	228
15000	0.597	0.001	248	1667	0	442
20000	0.554	0.001	139	2461	0	690
25000	0.523	0.001	73	3365	0	986
30000	0.5	0.001	49	4284	0	1292
35000	0.482	0.001	30	5031	0	1604
40000	0.473	0.001	20	5611	0	1900
45000	0.466	0.001	14	6121	0	2186
50000	0.459	0.001	6	6504	0	2457

4.3. Cross validation approaches

The grouping of observations into spatially distinct clusters for both the assessment of model extrapolation performance and the implementation of the spatial cross-validation approach was done using K-means clustering. The K-means algorithm aims to partition the data in a way that minimizes the total within cluster variation, typically expressed as the squared Euclidean distance between observations in each cluster (James et al., 2021). For clusters $C_1, \dots, C_K$ the optimization problem the algorithm attempts to solve can be defined as

$$\underset{C_1, \dots, C_K}{minimize} \left\{ \sum_{k=1}^{K} \frac{1}{|C_1|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} \left(x_{ij} - x_{i'j} \right)^2 \right\}$$

Where $|C_1|$ is the number of observations in cluster C_1 and p the number of variables used in the clustering. Unfortunately it is typically not possible to find an exact solution to this problem as the possible ways to partition the data grows exponentially with the number of observations (James et al., 2021). The K-means algorithm instead finds a reasonable approximation and is defined by James et al. (2021) as the following:

1. Randomly assign each observation to one of the K clusters
2. Repeat the following until cluster assignment does not change
    a. Find the centroids for each cluster
    b. Assign each observation to the nearest cluster centroid

To account for the fact that the random assignment of observations to clusters finds a local optimum, multiple random starts are typically used with the overall best clustering chosen

(James et al., 2021) and in this study the K-means algorithm was run with100 random starts. There are several methods for choosing the value for K that produces the optimum clustering of the data such as the elbow method (Schubert, 2023), silhouette score (Rousseeuw, 1987), and the gap statistic (Tibshirani et al., 2001). However, the use of K-means in this study was purely to group observations for spatial cross validation and as such $K = 5$ was chosen out of convenience and because larger values used in the cross-validation were computationally infeasible with the available resources.

To evaluate the out-of-sample-area extrapolation performance of the different modelling strategies the entire dataset was partitioned into five spatially independent clusters using K-means with each cluster being used as a test set in a leave-one-group-out cross-validation approach. To ensure maximum spatial isolation between the test and training sets (called outer test and outer train) a buffer zone of 100km was added to each of the test areas such that any observations falling within the buffer zone were excluded from model training. A visualization of the partitioning of the entire dataset into outer test/train splits with a 100km buffer zone applied to the outer test set is displayed in Figure 5.

Three different cross validation strategies were used during model training (see Figure 5). The first was random cross-validation where each of the outer train sets was randomly split into k folds with each fold being used as the validation set for models trained on the remaining folds (referred to as inner train and inner validation). The other two approaches were examples of spatial cross-validation in which the outer train data was first split into k clusters using K-means, and then leave-one-group-out cross validation was employed to train and validate the models. The difference between the two spatial cross-validation approaches was that one included a buffer zone of 25km around the inner validation sets and the other did not.

Values of five or 10 for have been recommended for cross-validation approaches (Hastie et al., 2009; Kuhn and Johnson, 2013). In this study k was set at five rather than 10 due to limits in the available computer memory.

The inner cross-validation approaches were used to tune the random forest models with each value for m_{try} being used with each of the inner folds and the optimal value being determined as the one that produced the lowest mean RMSE across all five folds. The best iteration of each model configuration was then fit to the entire outer train set and the extrapolation prediction performance was then assessed against the outer test set.

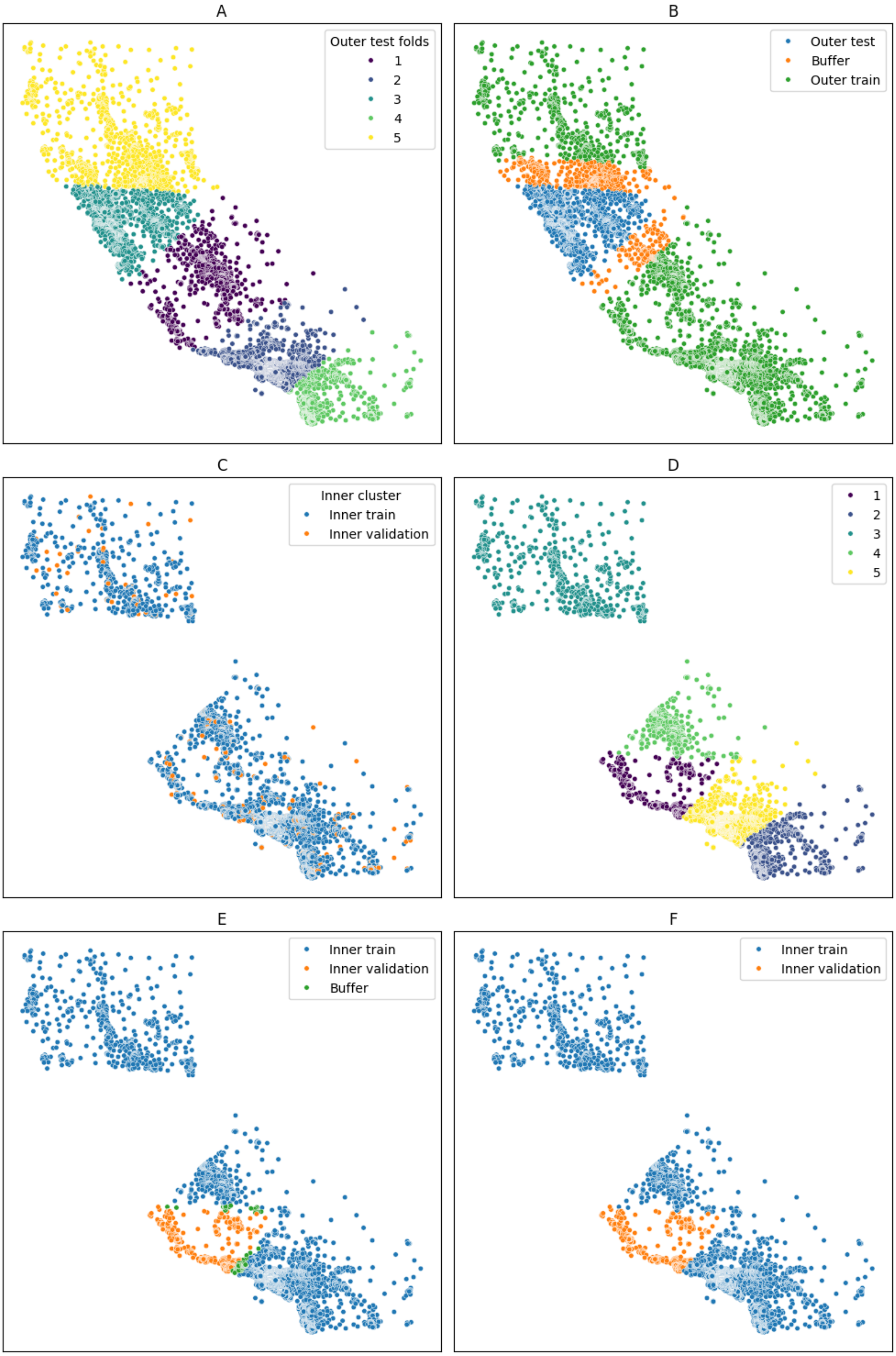


Figure 5. Plots of the cross-validation strategies used. ***A****. The split from K-means clustering creating the outer test folds.* ***B****. An example of an outer test set with a 100km buffer applied.* ***C****. An example fold from the random cross-validation approach indicating inner validation and train splits.* ***D****. the splits from K-means clustering of the outer train set for use with spatial cross-validation approaches.* ***E****. Example for the inner train and validation sets for spatial cross validation with buffer zone.* ***F****. Example for the inner train and validation sets for spatial cross validation without buffer zone.*

4.4. Cross-validation workflows

The workflows for each of the three cross-validation approaches are detailed below. As K-means clustering can produce slightly different clusters each time it is run, the random state was set prior to running the algorithm so that the cluster groups are consistent for both the outer and inner folds across approaches.

Random cross-validation workflow

1. Create spatially grouped clusters using K-means clustering (K =5)
2. Add relevant spatially lagged variables, to predictor set depending on the model used.
3. For each cluster create a buffer zone equal to 100km.
4. Using leave-one-group-out cross-validation loop through each of the outer folds and do the following.
    a. Split outer train set randomly into k folds (k=5)
    b. On each inner train set
        i. Center and scale predictors
        ii. Fit random forest model for six values of m_{try}
    c. Validate all models for all values of m_{try} using inner validation sets
    d. Average the RMSE across all 5 inner folds for each value of m_{try} and take the lowest mean RMSE as best tune.
    e. Use the best tune to fit random forest model to entire outer train set.
    f. Predict values for outer test and record RMSE
5. Average RMSE across all outer folds, weighted by test fold size.

Spatial cross-validation with buffer workflow

1. Create spatially grouped clusters using K-means clustering (K =5)
2. Add relevant spatially lagged variables, to predictor set depending on the model used.
3. For each cluster create a buffer zone equal to 100km.
4. Using leave-one-group-out cross-validation loop through each of the outer folds and do the following.
    a. Split outer train into spatially grouped clusters using k-means (k=5).
    b. Create 25kmbuffer around each inner validation set
    c. On each inner train set
        i. Center and scale predictors
        ii. Fit random forest model for six values of m_{try}
    d. Validate all models for all values of m_{try} using inner validation sets
    e. Average the RMSE across all 5 inner folds for each value of m_{try} and take the lowest mean RMSE as best tune.
    f. Use the best tune to fit random forest model to entire outer train set.
    g. Predict values for outer test and record RMSE
5. Average RMSE across all outer folds, weighted by test fold size.

Spatial cross-validation without buffer workflow

1. Create spatially grouped clusters using K-means clustering (K =5)
2. Add relevant spatially lagged variables, to predictor set depending on the model used.
3. For each cluster create a buffer zone equal to 100km.
4. Using leave-one-group-out cross-validation loop through each of the outer folds and do the following.
    a. Split outer train into spatially grouped clusters using k-means (k=5).
    b. On each inner train set
        i. Center and scale predictors
        ii. Fit random forest model for six values of m_{try}
    c. Validate all models for all values of m_{try} using inner validation sets
    d. Average the RMSE across all 5 inner folds for each value of m_{try} and take the lowest mean RMSE as best tune.
    e. Use the best tune to fit random forest model to entire outer train set.
    f. Predict values for outer test and record RMSE
5. Average RMSE across all outer folds, weighted by test fold size.

4.5. Model evaluation

The predictive performance for all models was evaluated using RMSE. The mean RMSE for both the validation and test predictions was weighted by fold size, as the K-means clustering creates unequal cluster sizes. The weighted mean $RMSE$ for the test folds is defined as

$$RMSE_{mean}^{test} = \sum_{k=1}^{K} \frac{n_k^{test}}{n^{test}} RMSE_k^{test}$$

where n^{test} is the total number of observations across all test folds, $n_k^{test}$is the number of observations in test fold k and $RMSE_k^{test}$ the root mean squared error for fold k defined as

$$RMSE_k^{test} = \sqrt{\frac{1}{n_k^{test}} \sum_{i=1}^{n_k^{test}} (y_i - \hat{y}_i)^2}$$

With y_i and $\hat{y}_i$ the i^{th} observed and predicted values for the response variable (James et al., 2021). Note, the validation mean $RMSE_{mean}^{valid}$ is calculated in the same manner using the validation fold sizes as weighting. The standard error for the weighted $RMSE_{mean}^{valid}$ and $RMSE_{mean}^{test}$ were both calculated in the following manner (“Weighted arithmetic mean,” 2024).

$$se^{test} = \sqrt{\frac{\sum_{k=1}^{K} n_k^{test} (RMSE_k^{test} - RMSE_{mean}^{test})^2}{\sum_{k=1}^{K} n_k^{test} - 1}}$$

It should be noted that this is not a truly accurate representation of the cross-validation standard error as the cross validation folds are not strictly independent, but it provides a useful approximation (*Statistical Learning*, 2022).

As the aim of the project was to evaluate the out of sample area extrapolation performance of the different cross-validation approaches the difference in test and validation $RMSE$ for each outer fold, weighted by the size of the test fold, was calculated for each model as

$$Diff_{mean} = \sum_{k=1}^{K} \frac{n_k^{test}}{n^{test}} Diff_k$$

with

$$Diff_k = RMSE_k^{test} - RMSE_k^{valid}$$

The standard error for $Diff_{mean}$ was calculated as the pooled standard error for $RMSE_k^{test}$ and $RMSE_k^{valid}$ and calculated as

$$se^{diff} = \sqrt{\frac{\left(se^{test^2} + se^{valid^2}\right)}{K}}$$

p-values for the null hypothesis that the mean difference in test – validation RMSE was equal to zero were obtained using independent samples t-test's with $K + K - 2$ degrees of freedom, for all models. One of the assumptions of the independent samples t-test is that the data are normally distributed and so a Shapiro-Wilks test was conducted to test this assumption for the RMSE values across the validation and test data (see Appendix 1. Table 1). Although there was evidence of non-normality for few instances, it was decided to persist with the t-test as it has greater power than the non-parametric alternatives (Lee, 2014).

To account for the fact that multiple models were being evaluated and thus the likelihood of a type-I error occurring increased, p-values were adjusted using the Holm's step-down procedure to control the family wise error rate (FWER) (James et al., 2021). The FMER is the probability of at least one type-I error occurring when testing m null hypotheses $H_{01}, \dots, H_{0m}$ defined as

$$FWER = \Pr\,(V \geq 1)$$

Where V is the number of null hypotheses that were rejected incorrectly. Holm's step-down procedure controls the $FWER$ at a level of α using the following algorithm

1. Determine value for α
2. Calculate p-values $p_1, \dots, p_m$ for all m null hypotheses $H_{01}, \dots, H_{0m}$
3. Arrange the p-values in ascending order such that $p_{(1)}, \leq p_{(2)} \leq \cdots \leq p_{(m)}$
4. Let $j = min\{i : p_{(i)} > \alpha/(m + 1 - i)\}$
5. Reject all H_{0i} for $p_i < p_{(j)}$

In this study the value for α was set to 0.05.

The differences in the mean RMSE from both validation and test between the three cross-validation approaches using the same model specification (e.g. random_lag5_predictors vs spatial_lag5_predictors) were also calculated using the above procedure.

4.6. Software

The random forest model and model cross-validation strategies were implemented with R Statistical Software (R Core Team, 2023) using the following packages; tidyverse (Wickham et al., 2019), Tidymodels (Kuhn and Wickham, 2020), randomForest (Liaw and Wiener, 2002), spatialsample (Mahoney et al., 2023), sf (Pebesma and Bivand, 2023)

Exploratory data analysis, feature engineering, data visualizations, calculation of spatial autocorrelation metrics and creation of lagged variables was done with the Python programming language (van Rossum, 1995), version 3.10.12, using the following packages; Pandas (McKinnney, 2010), Numpy (Harris et al., 2020), Matplotlib (Hunter, 2007), seaborn (Waskom, 2021), PySAL (Rey and Anselin, 2007), SciKit-Gstat (Mälicke, 2022), geopandas (Joris Van den Bossche et al., 2024), Shapely (Gillies et al., 2022), splot (Marcilio Mendonca, 2015)

4.7. Summary of modelling approaches

The three cross validation strategies using combinations of either no lagged variables, lagged response, or lagged predictors and created from the two distance band weights matrices in total resulted in 14 models to evaluate and are detailed in Table 4.

Table 4. Specification of modelling approaches evaluated

	Cross-validation method			**Lagged variables included**				
Model	**Random CV**	**Spatial CV**	**Spatial CV with buffer**	**None**	**Lag 5km response**	**Lag 5km predictors**	**Lag 20km response**	**Lag 20km predictors**
random_no_lag	x			x				
random_lag5_response	x				x			
random_lag5_predictors	x					x		
random_lag20_response	x						x	
random_lag20_predictors	x							x
spatial_no_lag		x		x				
spatial_lag5_response		x			x			
spatial_lag5_predictors		x				x		
spatial_lag20_response		x					x	
spatial_lag20_predictors		x						x
spatial_buff_lag5_response			x		x			
spatial_buff_lag5_predictors			x			x		
spatial_buff_lag20_response			x				x	
spatial_buff_lag20_predictors			x					x

5. Results

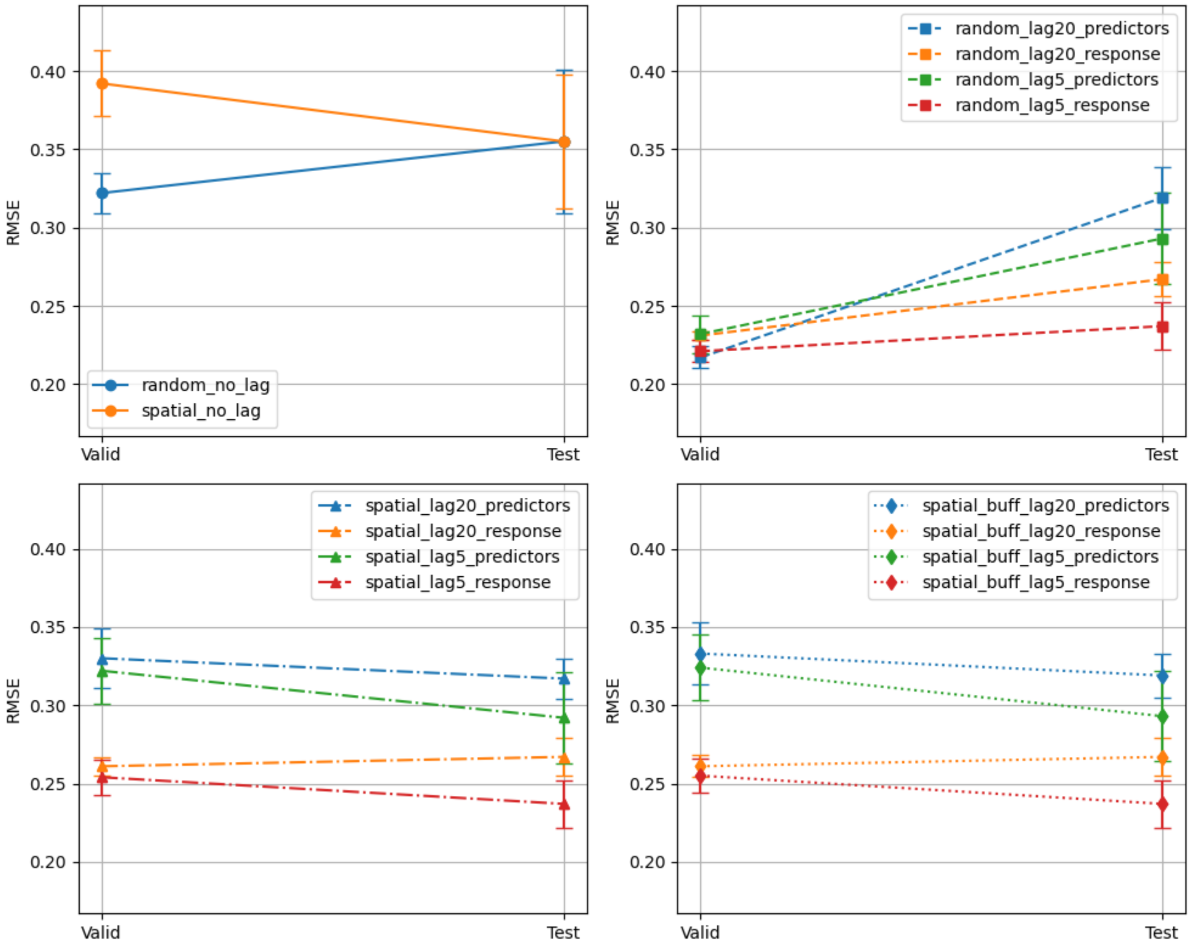


Figure 6. Weighted mean RMSE with standard error for validation and test for all models.

The validation and test weighted mean RMSE with standard error for all models are displayed in Figure 6. There was a general trend in which the random cross-validation approaches tended to have worse performance on the test compared to validation data and the spatial cross-validation approaches had slightly improved performance on the test compared to validation data.

The differences in weighted mean RMSE between testing and validation, with standard error intervals, for all models are displayed in Figure 7. There was only evidence to reject the null hypothesis of no difference between the test and validation mean RMSE values for three models and these were random_lag20_predictors, random_lag20_response and random_lag5_predictors. The largest difference in test – validation mean RMSE for these three models was observed in the random_lag20_predictors model.

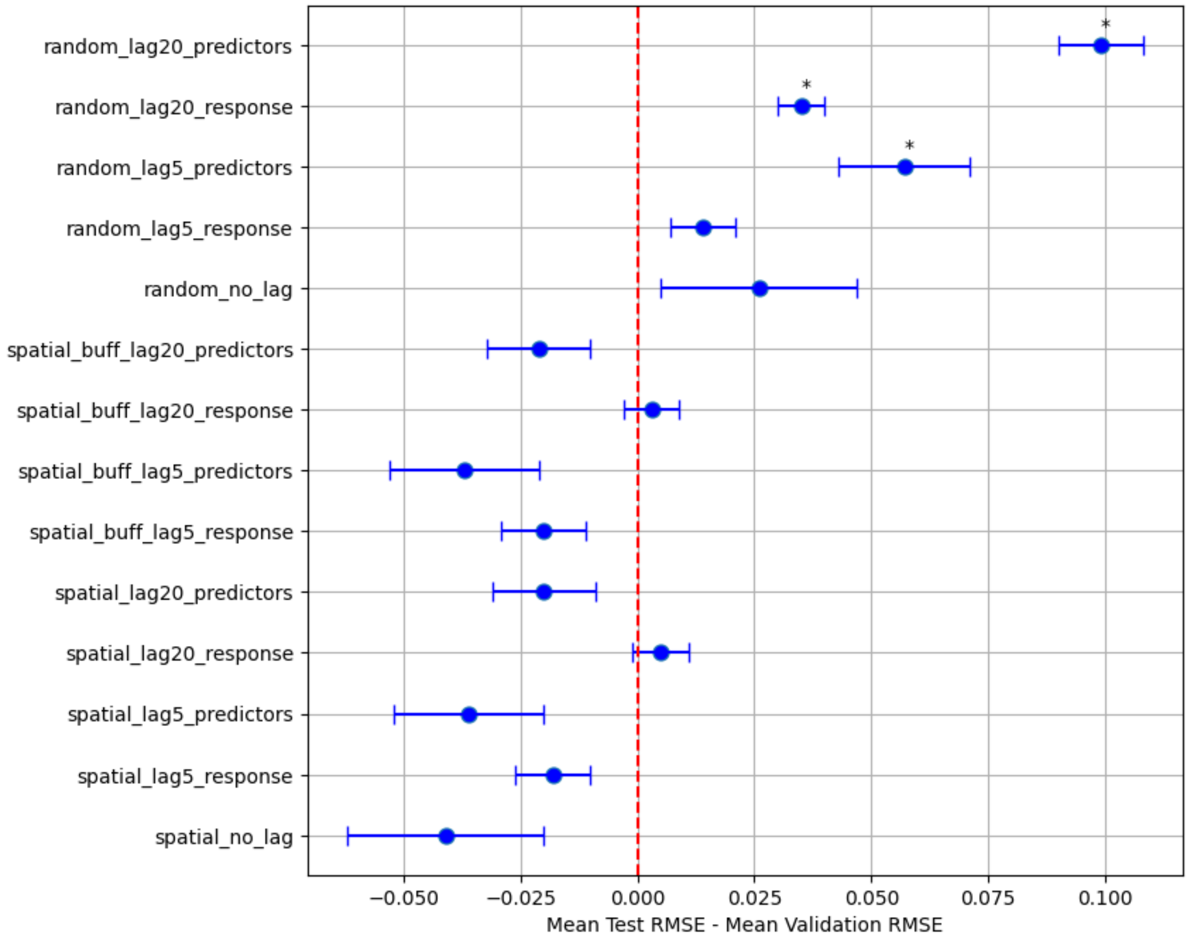


Figure 7. Difference in mean test and mean validation RMSE with standard error for all models. An Asterix indicates models where the adjusted p-value was below the threshold value of α = 0.05.

Figure 8 displays the difference in mean validation RMSE between the cross-validation approaches matched by the type of lagged variables added to the model. In all cases the spatial and spatial with buffer cross-validation approaches had significantly higher mean validation RMSE scores compared to the random approach, with modified p-values below the threshold value for α of 0.05. There was no significant difference in mean validation RMSE between the spatial and spatial with buffer approaches for any combination of lagged variables.

When comparing the mean test RMSE between cross-validation approaches matched by the type of lagged variable added (see Figure 9) there was no significant difference between any cross-validation approaches for any combination of lagged variables.

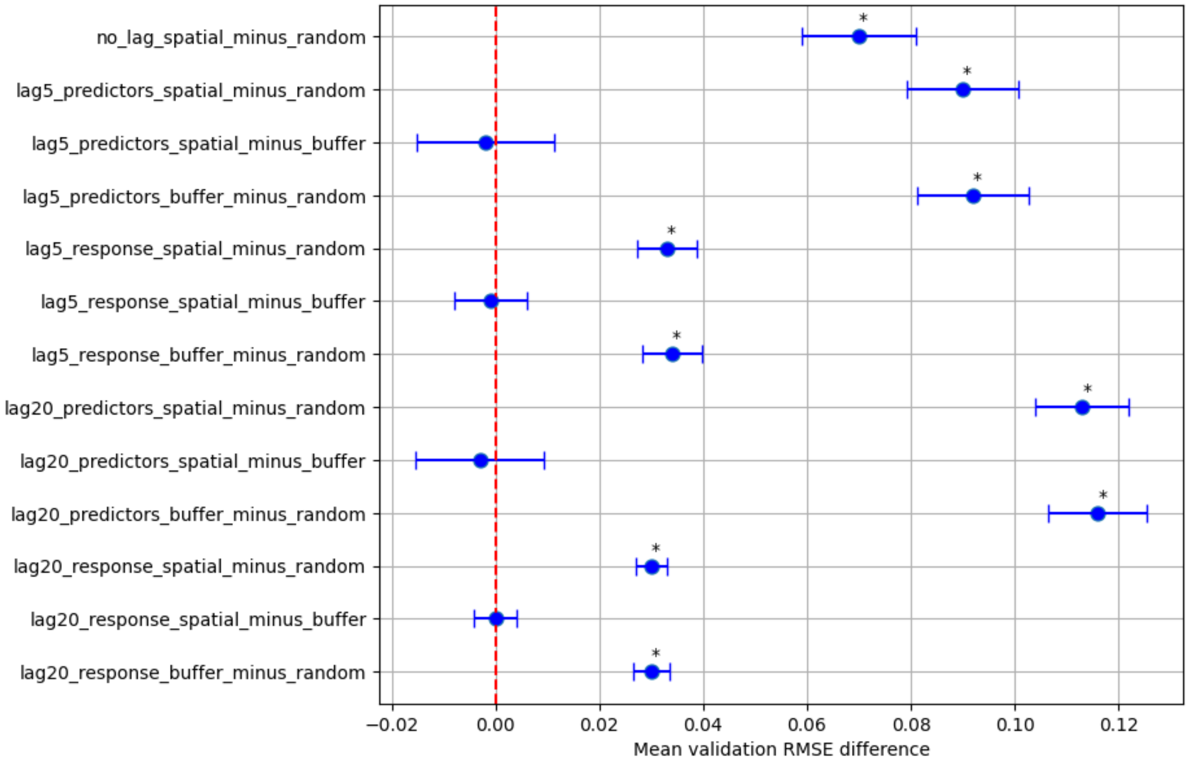


Figure 8. Difference in weighted mean validation RMSE with standard error, between the three cross-validation approaches, with and without lagged variables. An Asterix indicates models where the adjusted p-value was below the threshold value of α = 0.05.

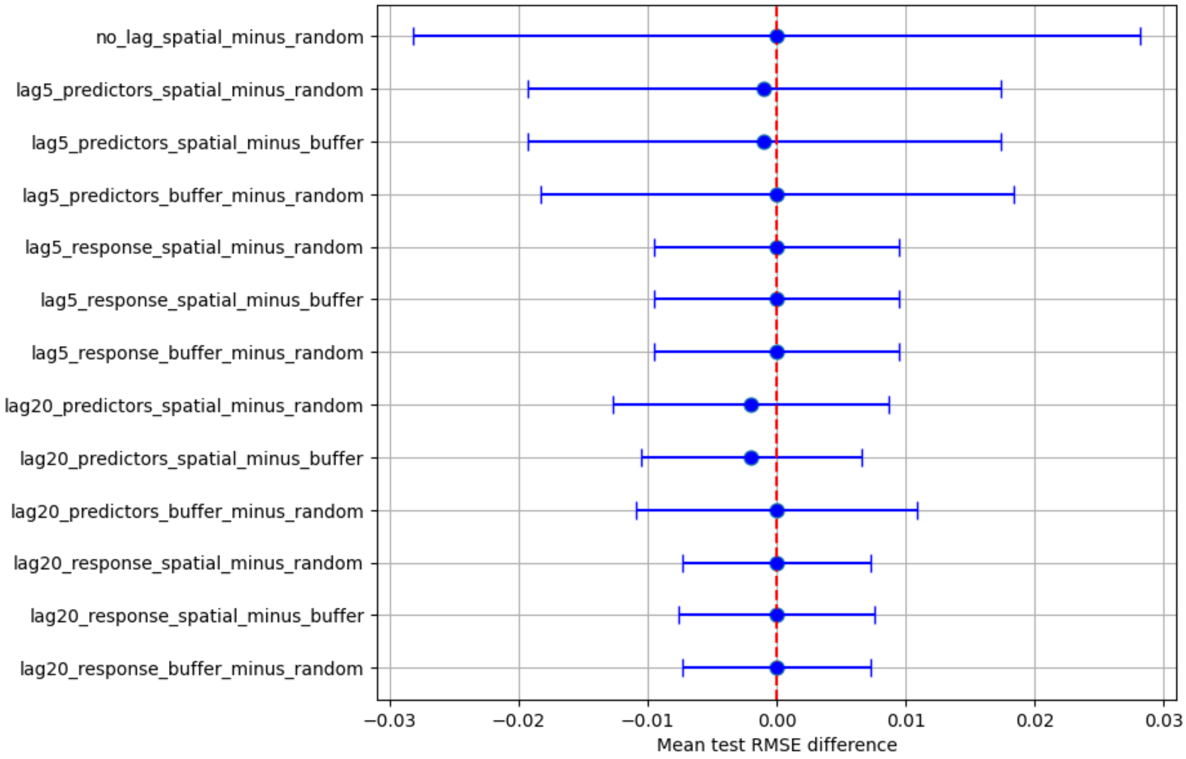


Figure 9. Difference in weighted mean test RMSE with standard error, between the three cross-validation approaches, with and without lagged variables. No models had an adjusted p-value that was below the threshold value of α = 0.05.

The mean relative variable importance scores, averaged across the outer test folds, for the random_lag20_predictors, random_lag5_predictors, random_lag20_response and random_lag5_response models are displayed in Table 5. The scores for these four models are displayed as the risk of data leakage from lagged variables is greatest when included with the random cross-validation approaches. For each of the four models represented in Table 5 lagged variables had the highest importance score which indicates that information from neighbouring locations was used heavily in model fitting.

Table 5. Relative variable importance averaged across the outer test folds for the random cross-validation models that included lagged variables.

random_lag20_predictors	
Variable	**Relative importance**
log_medianIncome_lag20	100%
log_medianIncome	95%
housingMedianAge_lag20	31%
log_averageRooms	25%
log_popPerHouse	25%
log_averageRooms_lag20	21%
log_population_lag20	21%
log_averageBedrooms_lag20	19%
log_popPerHouse_lag20	17%
housingMedianAge	7%
log_averageBedrooms	6%
log_population	6%

random_lag5_predictors	
Variable	**Relative importance**
log_medianIncome_lag5	100%
log_medianIncome	78%
housingMedianAge_lag5	20%
log_popPerHouse_lag5	19%
log_averageRooms_lag5	18%
log_averageRooms	15%
log_popPerHouse	15%
log_population_lag5	8%
log_averageBedrooms_lag5	7%
housingMedianAge	6%
log_population	6%
log_averageBedrooms	5%

random_lag20_response	
Variable	**Relative importance**
log_medianHouseValue_lag20	100%
log_medianIncome	56%
log_popPerHouse	18%
log_averageRooms	16%
housingMedianAge	7%
log_averageBedrooms	6%
log_population	6%

random_lag5_response	
Variable	**Relative importance**
log_medianHouseValue_lag5	100%
log_medianIncome	42%
log_averageRooms	12%
log_popPerHouse	10%
housingMedianAge	5%
log_averageBedrooms	5%
log_population	4%

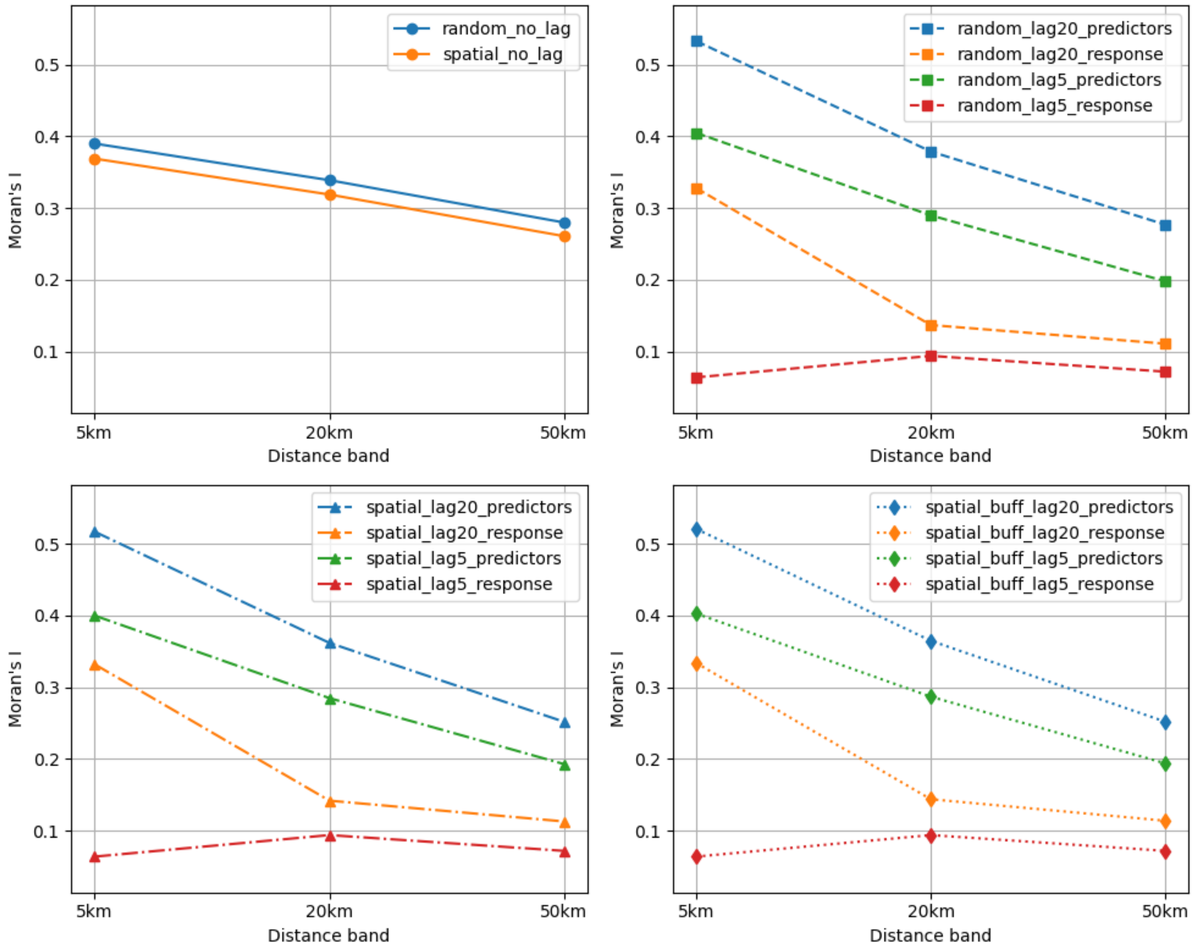


Figure 10. Moran's I for model residuals from predictions on test folds calculated at 5km, 20km, and 50km distance bands. All values have simulated p-values of 0.001.

Figure 10 displays the Moran's I for test prediction residuals for distance bands of 5km, 20km and 50km for all models. In general, the cross-validation approach did not have a large effect on the pattern of residual spatial autocorrelation. The biggest impact was associated with the inclusion (or not) of lagged variables. Models that included the response variable lagged at 5km regardless of the cross-validation approach had almost no remaining spatial autocorrelation in the residuals calculated at any distance band (all had Moran's I less than 0.1). Including the response variable lagged at 20km resulted in the second lowest residual Moran's I values with a similar pattern across the cross-validation approaches in which at 5km the value was approximately 0.33, at 20km there was a large drop to approximately 0.14 and at 50km had dropped further to approximately 0.11. The models that included predictors lagged at 5km had Moran's I values of approximately 0.4, 0.38 and 0.19 at 5km, 20km and 50km distances respectively. Including predictors lagged at 20km produced the highest residual Moran's I scores of all the models with values of approximately 0.52 at 5km, between 0.36 and 0.38 at 20km, and between 0.25 and 0.28 at 50km. Not including lagged variables resulted in Moran's I values of 0.37 and 0.39 at 5km, 0.32 and 0.34 at 20km and 0.26 and 0.28 at 50km for the spatial and random cross-validation approaches respectively.

6. Discussion

Many studies have shown that when spatial autocorrelation exists within the response variable, random cross-validation leads to overly optimistic results with a decline in performance when predictions are made out of the spatial area used to train the model (Mahoney et al., 2023; Roberts et al., 2017; Schratz et al., 2019; Wang et al., 2023). The results of this study are in line with these findings in that all the random models tended to have a decline in performance during testing, although the difference was only significant in three of the five random models. Further evidence of the random cross-validation approaches propensity to produce overly optimistic results in this setting is provided by the comparison in validation performance of the random approach with the two spatial approaches, matched for the type of lagged variables included in the model. In all cases the random approach had significantly lower mean validation RMSE when compared with both spatial approaches, while there was no significant difference in mean test RMSE between any of the approaches. Including a buffer zone in the spatial cross-validation process has been recommended as an additional measure to reduce optimistically biased results (Mahoney et al., 2023). In this study there was no significant difference in the mean validation or mean test RMSE between the spatial cross-validation method with a buffer included and the one without suggesting the creation of cross-validation folds via spatial clustering was sufficient to prevent overfitting.

The two spatial cross-validation approaches tended to have slightly better performance when predicting against the test sets compared to validation performance, however the differences were not significant for any of the nine spatial models. The tendency for better performance is somewhat surprising considering that cross-validation based on spatially clustered folds was used to obtain validation and tests scores. One possible factor that could explain the trend was the fact that less data was used to fit and evaluate the models during validation compared with testing.

As has been shown in other studies (Kiely and Bastian, 2020; Liu et al., 2022; Pawley et al., 2024) the addition of spatially lagged variables improved performance of the models compared to models without them and. However, one of the main aims of this study was to evaluate whether the inclusion of spatially lagged variables could lead to data leakage during model training. When using the random cross-validation approach the only three models that had a significant decline in performance from validation to test were those that included lagged variables. The greatest decline was found with the addition of predictors lagged using the larger distance band of 20km followed by predictors lagged at 5km, and finally the response lagged at 20km. This is consistent with what would be expected if data leakage was occurring as adding lagged predictors adds more information from neighbouring locations than adding the lagged response (there are five predictors to one response variable), and the larger 20km distance band included on average 10 times the number of neighbours compared to the 5km distance band. The fact that lagged variables had the highest variable importance scores in each of these three models gives further support for data leakage potentially occurring in these three models. In principle it is possible for data leakage to also occur in models with lagged variables using the spatial-cross-validation approach without buffer zones around the validation folds. No evidence of this occurring was found in the present study as there was no significant decline in performance from validation to testing for any of the spatial models, and no significant difference in validation or test mean RMSE between the two spatial approaches for any of the model configurations. Two factors that could have impacted this are that the K-means algorithm tends to create folds with most of the observations closer to the center and fewer along the fold periphery. A second factor is that that due to the number of cross-validation folds being small (k = 5) the fold sizes were quite large so that most observations in the test set were not within the threshold lag distance of observations in the validation set.

The cross-validation strategy used had little impact on the spatial autocorrelation of the test prediction residuals, but the inclusion of lagged variables did. For all cross-validation approaches the inclusion of the response variable lagged at 5km had the greatest impact, with an almost complete reduction in spatial autocorrelation. Interestingly, including predictors lagged at 20km resulted in greater residual spatial autocorrelation than models with no lagged variables included, which indicates the importance of getting the correct specification for any lagged variables included in the modeling process.

7. Limitations

The standard errors for the validation vs test differences for many of the models were large, especially for the random and spatial approaches with no lagged variables. Running the cross-validation strategies for multiple iterations of each would reduce the standard error and potentially show more significant differences between models. Furthermore, the t-test used to calculate the p-values assumes the validation and test fold RMSE values are normally distributed, and this assumption is not valid for all models tested. If multiple repeats of each cross-validation approach were used bootstrap confidence intervals could be calculated which do not have assumptions about the distributions of the data. Unfortunately using multiple repeats of the cross-validation approaches was not possible in this study due to time and resource constraints.

The spatial autocorrelation in the response variable used in this study shows a clear east to west trend with higher median house prices located along the coast and relatively lower prices inland (see Figure 2). The K-means clustering algorithm split the test folds in a manner such that the total sample area was divided from north to south (see Figure 5). As a result of this each fold contained observations located both on the coast and inland which means the individual test folds may not have been that dissimilar from each other, and therefore not the best test of model out of sample area extrapolation performance. Splitting the sample space into more clusters or using a different approach to create the outer test cross-validation folds such as grouping the observations by region, as was done in the study by Wang et, al. (2023) may have provided a better extrapolation test.

The lagged variables used in this study were created using a weights matrix constructed from distance bands and this specification meant a large number of observations were counted as neighbours. Other specifications of the spatial weights matrix that include fewer neighbours could result in any data leakage being less of a problem.

8. Conclusions

This study aimed to investigate the effect of different cross-validation techniques in combination with spatially lagged variables on the extrapolation performance of random forest models when spatial autocorrelation was present in the response variable. As expected, models trained using the random cross-validation approach tended to have overly optimistic performance during validation with subsequently higher prediction error when used on the test folds, but this was only significant in models that included lagged variables. This latter point, and the fact that the model with the most lagged variables created using the largest lag distance and trained using random cross-validation had the highest test minus validation RMSE difference, provides some support for the data leakage occurring from lagged variables when random cross validation is used. While spatial cross spatial cross-validation based on spatially grouped clusters reduces the potential for data leakage occurring from lagged predictors, it is still possible in principle unless a buffer zone is used to isolate spatially lagged validation observations from the training process. In this study no evidence of overfitting was found for either of the spatial cross-validation approaches suggesting that if data leakage did occur during the spatial method without a buffer, it was not large enough to have a detectable effect. Several limitations exist with this study and as such it represents a preliminary investigation showing data leakage from spatially lagged variables potentially contributing to poor extrapolation performance in machine learning models when combined with random cross-validation. Further research is required to investigate this in more detail and to identify the specific conditions under which data leakage can occur and the magnitude of its effect.

9. Recommendations

Further research using a simulated dataset similar that used by Mahoney et al. (2023) and Roberts et al. (2017) would allow for a more controlled investigation into this topic. Controlling for trend in the pattern of spatial-autocorrelation and understanding the underlying process used to generate the spatial autocorrelation would allow for a specification of the spatial weights matrix that matched the spatial dependence in the data. Using a synthetic dataset would also allow the exact relationship between the response variable and the predictors to be known which would aid in determining which lagged predictors to include in the modelling process. Performing multiple repeats of the different cross-validation approaches would provide more information about the distribution of values for the mean RMSE for each approach and allow more accurate inference about the impact of lagged variable on extrapolation performance. It would also be beneficial to quantify the number of observations from validation sets that were included in the creation of lagged variables used during model training to gain an understanding of the magnitude of any potential data leakage.

10. References

Anselin, L., 2021. Spatial Models in Econometric Research. https://doi.org/10.13140/RG.2.2.26447.20641

Anselin, L., 2002. Under the hood Issues in the specification and interpretation of spatial regression models. Agricultural Economics 27, 247–267. https://doi.org/10.1111/j.1574-0862.2002.tb00120.x

Behrens, T., Schmidt, K., Viscarra Rossel, R.A., Gries, P., Scholten, T., MacMillan, R.A., 2018. Spatial modelling with Euclidean distance fields and machine learning. European Journal of Soil Science 69, 757–770. https://doi.org/10.1111/ejss.12687

Bivand, R., Pebesma, E., Gomez-Rubio, V., 2008. Applied Spatial Data Analysis with R. Springer, New York, NY. https://doi.org/10.1007/978-0-387-78171-6

Breiman, L., 2001. Random Forests. Machine Learning 45, 5–32. https://doi.org/10.1023/A:1010933404324

Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1984. Classification And Regression Trees, 1st ed. Routledge. https://doi.org/10.1201/9781315139470

Brenning, A., 2012. Spatial cross-validation and bootstrap for the assessment of prediction rules in remote sensing: The R package sperrorest, in: 2012 IEEE International Geoscience and Remote Sensing Symposium. Presented at the IGARSS 2012 - 2012 IEEE International Geoscience and Remote Sensing Symposium, IEEE, Munich, Germany, pp. 5372–5375. https://doi.org/10.1109/IGARSS.2012.6352393

Cai, L., Kreft, H., Taylor, A., Denelle, P., Schrader, J., Essl, F., van Kleunen, M., Pergl, J., Pyšek, P., Stein, A., Winter, M., Barcelona, J.F., Fuentes, N., Inderjit, Karger, D.N., Kartesz, J., Kuprijanov, A., Nishino, M., Nickrent, D., Nowak, A., Patzelt, A., Pelser, P.B., Singh, P., Wieringa, J.J., Weigelt, P., 2023. Global models and predictions of plant diversity based on advanced machine learning techniques. New Phytologist 237, 1432–1445. https://doi.org/10.1111/nph.18533

Cliff, A.D., Ord, K., 1970. Spatial Autocorrelation: A Review of Existing and New Measures with Applications. Economic Geography.

Credit, K., 2022. Spatial Models or Random Forest? Evaluating the Use of Spatially Explicit Machine Learning Methods to Predict Employment Density around New Transit Stations in Los Angeles. Geographical Analysis 54, 58–83. https://doi.org/10.1111/gean.12273

Cressie, N., 1990. The origins of kriging. Math Geol 22, 239–252. https://doi.org/10.1007/BF00889887

Dormann, C., McPherson, J., Araújo, M., Bivand, R., Bolliger, J., Carl, G., G. Davies, R., Hirzel, A., Jetz, W., Daniel Kissling, W., Kühn, I., Ohlemüller, R., R. Peres-Neto, P., Reineking, B., Schröder, B., M. Schurr, F., Wilson, R., 2007. Methods to account for spatial autocorrelation in the analysis of species distributional data: a review. Ecography 30, 609–628. https://doi.org/10.1111/j.2007.0906-7590.05171.x

Elhorst, J.P., 2014. Spatial Econometrics: From Cross-Sectional Data to Spatial Panels, SpringerBriefs in Regional Science. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-40340-8

Gillies, S., van der Wel, C., Van den Bossche, J., Taves, M.W., Arnott, J., Ward, B.C., Others, 2022. Shapely. https://doi.org/10.5281/ZENODO.7428463

Griffith, D.A., 2009. Spatial Autocorrelation, in: International Encyclopedia of Human Geography. Elsevier, pp. 308–316. https://doi.org/10.1016/B978-008044910-4.00522-8

Harris, C.R., Millman, K.J., Van Der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., Van Kerkwijk, M.H., Brett, M., Haldane, A., Del Río, J.F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E., 2020. Array programming with NumPy. Nature 585, 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning, Springer Series in Statistics. Springer New York, New York, NY. https://doi.org/10.1007/978-0-387-84858-7

Hengl, T., Nussbaum, M., Wright, M.N., Heuvelink, G.B.M., Gräler, B., 2018. Random forest as a generic framework for predictive modeling of spatial and spatio-temporal variables. PeerJ 6, e5518. https://doi.org/10.7717/peerj.5518

Hu, X., Belle, J.H., Meng, X., Wildani, A., Waller, L.A., Strickland, M.J., Liu, Y., 2017. Estimating PM2.5 Concentrations in the Conterminous United States Using the Random Forest Approach. Environ. Sci. Technol. 51, 6936–6944. https://doi.org/10.1021/acs.est.7b01210

Hunter, J.D., 2007. Matplotlib: A 2D Graphics Environment. Comput. Sci. Eng. 9, 90–95. https://doi.org/10.1109/MCSE.2007.55

James, G., Witten, D., Hastie, T., Tibshirani, R., 2021. An Introduction to Statistical Learning: with Applications in R, Springer Texts in Statistics. Springer US, New York, NY. https://doi.org/10.1007/978-1-0716-1418-1

Joris Van den Bossche, Kelsey Jordahl, Martin Fleischmann, Matt Richards, James McBride, Jacob Wasserman, Adrian Garcia Badaracco, Alan D. Snow, Brendan Ward, Jeff Tratner, Jeffrey Gerard, Matthew Perry, Carson Farmer, Geir Arne Hjelle, Mike Taves, Ewout ter Hoeven, Micah Cochran, Ray Bell, rraymondgh, Matt Bartos, Pieter Roggemans, Lucas Culbertson, Giacomo Caria, Nick Eubank, sangarshanan, John Flavin, Sergio Rey, James Gardiner, Kaushik, 2024. geopandas/geopandas: v0.14.4. https://doi.org/10.5281/ZENODO.11080352

Kämäräinen, M., Tuovinen, J.-P., Kulmala, M., Mammarella, I., Aalto, J., Vekuri, H., Lohila, A., Lintunen, A., 2023. Spatiotemporal lagging of predictors improves machine learning estimates of atmosphere–forest CO_2 exchange. Biogeosciences 20, 897–909. https://doi.org/10.5194/bg-20-897-2023

Kapoor, S., Narayanan, A., 2023. Leakage and the reproducibility crisis in machine-learning-based science. Patterns (N Y) 4, 100804. https://doi.org/10.1016/j.patter.2023.100804

Kelley Pace, R., Barry, R., 1997. Sparse spatial autoregressions. Statistics & Probability Letters 33, 291–297. https://doi.org/10.1016/S0167-7152(96)00140-X

Kiely, T.J., Bastian, N.D., 2020. The spatially conscious machine learning model. Statistical Analysis and Data Mining: The ASA Data Science Journal 13, 31–49. https://doi.org/10.1002/sam.11440

Kuhn, M., Johnson, K., 2013. Applied Predictive Modeling. Springer, New York, NY. https://doi.org/10.1007/978-1-4614-6849-3

Kuhn, M., Wickham, H., 2020. Tidymodels: a collection of packages for modeling and machine learning using tidyverse principles.

Lee, C.-H., Greiner, R., Schmidt, M., 2005. Support Vector Random Fields for Spatial Classification, in: Jorge, A.M., Torgo, L., Brazdil, P., Camacho, R., Gama, J. (Eds.), Knowledge Discovery in Databases: PKDD 2005. Springer, Berlin, Heidelberg, pp. 121–132. https://doi.org/10.1007/11564126_16

Lee, H., 2014. Foundations of Applied Statistical Methods. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-02402-8

Legendre, P., Fortin, M.J., 1989. Spatial pattern and ecological analysis. Vegetatio 80, 107–138. https://doi.org/10.1007/BF00048036

Li, J., 2022. Spatial Predictive Modeling with R, 1st ed. Chapman and Hall/CRC, Boca Raton. https://doi.org/10.1201/9781003091776

Liaw, A., Wiener, M., 2002. Classification and Regression by randomForest. R news 2, 18–22.

Liu, X., Kounadi, O., Zurita-Milla, R., 2022. Incorporating Spatial Autocorrelation in Machine Learning Models Using Spatial Lag and Eigenvector Spatial Filtering Features. ISPRS International Journal of Geo-Information 11, 242. https://doi.org/10.3390/ijgi11040242

Mahoney, M.J., Johnson, L.K., Silge, J., Frick, H., Kuhn, M., Beier, C.M., 2023. Assessing the performance of spatial cross-validation approaches for models of spatially structured data. https://doi.org/10.48550/arXiv.2303.07334

Mälicke, M., 2022. SciKit-GStat 1.0: a SciPy-flavored geostatistical variogram estimation toolbox written in Python. Geoscientific Model Development 15, 2505–2532. https://doi.org/10.5194/gmd-15-2505-2022

Marcilio Mendonca, 2015. Splot. https://doi.org/10.5281/ZENODO.322478

Matheron, G., 1963. Principles of geostatistics. Economic Geology 58, 1246–1266. https://doi.org/10.2113/gsecongeo.58.8.1246

McKinnney, W., 2010. Data structures for statistical computing in python, in: Proc.of the 9th Python in Science Conference. Austin, Texas, pp. 51–56.

Melo, O.O., Mateu, J., Melo, C.E., 2016. Spatial generalised linear mixed models based on distances. Stat Methods Med Res 25, 2138–2160. https://doi.org/10.1177/0962280213515792

Meyer, H., Pebesma, E., 2022. Machine learning-based global maps of ecological variables and the challenge of assessing them. Nat Commun 13, 2208. https://doi.org/10.1038/s41467-022-29838-9

Meyer, H., Reudenbach, C., Wöllauer, S., Nauss, T., 2019. Importance of spatial predictor variable selection in machine learning applications – Moving from data reproduction to spatial prediction. Ecological Modelling 411, 108815. https://doi.org/10.1016/j.ecolmodel.2019.108815

Miller, J., Franklin, J., Aspinall, R., 2007. Incorporating spatial dependence in predictive vegetation models. Ecological Modelling 202, 225–242. https://doi.org/10.1016/j.ecolmodel.2006.12.012

Moran, P.A.P., 1948. The Interpretation of Statistical Maps. Journal of the Royal Statistical Society. Series B (Methodological) 10, 243–251.

Nikparvar, B., Thill, J.-C., 2021. Machine Learning of Spatial Data. ISPRS International Journal of Geo-Information 10, 600. https://doi.org/10.3390/ijgi10090600

Pawley, S.M., Atkinson, L., Utting, D.J., Hartman, G.M.D., Atkinson, N., 2024. Evaluating spatially enabled machine learning approaches to depth to bedrock mapping, Alberta, Canada. PLOS ONE 19, e0296881. https://doi.org/10.1371/journal.pone.0296881

Pebesma, E., Bivand, R., 2023. Spatial Data Science: With Applications in R, 1st ed. Chapman and Hall/CRC, New York. https://doi.org/10.1201/9780429459016

Pohjankukka, J., Pahikkala, T., Nevalainen, P., Heikkonen, J., 2017. Estimating the prediction performance of spatial models via spatial k-fold cross validation. International Journal of Geographical Information Science 31, 2001–2019. https://doi.org/10.1080/13658816.2017.1346255

Probst, P., Boulesteix, A.-L., Bischl, B., 2019. Tunability: Importance of Hyperparameters of Machine Learning Algorithms. Journal of Machine Learning Research 20, 1–32.

Quinlan, J.R., 1992. Learning with continuous classes, in: Proceedings of Australian Joint Conference on Artificial Intelligence. World Scientific, Hobart, pp. 343–348.

Quiñones, S., Goyal, A., Ahmed, Z.U., 2021. Geographically weighted machine learning model for untangling spatial heterogeneity of type 2 diabetes mellitus (T2D) prevalence in the USA. Sci Rep 11, 6955. https://doi.org/10.1038/s41598-021-85381-5

R Core Team, 2023. R: A language and environment for statistical computing.

Rey, S.J., Anselin, L., 2007. PySAL: A Python library of spatial analytical methods. Review of Regional Studies 37, 5–27.

Rey, S.J., Arribas-Bel, D., Wolf, L.J., 2023. Geographic data science with Python, Chapman & Hall/CRC texts in statistical science. CRC Press, Taylor & Francis Group, Boca Raton.

Roberts, D.R., Bahn, V., Ciuti, S., Boyce, M.S., Elith, J., Guillera-Arroita, G., Hauenstein, S., Lahoz-Monfort, J.J., Schröder, B., Thuiller, W., Warton, D.I., Wintle, B.A., Hartig, F., Dormann, C.F., 2017. Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. Ecography 40, 913–929. https://doi.org/10.1111/ecog.02881

Rousseeuw, P.J., 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 20, 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

Sabek, I., Mokbel, M.F., 2020. Machine Learning Meets Big Spatial Data, in: 2020 IEEE 36th International Conference on Data Engineering (ICDE). Presented at the 2020 IEEE 36th International Conference on Data Engineering (ICDE), pp. 1782–1785. https://doi.org/10.1109/ICDE48307.2020.00169

Salazar, J.J., Garland, L., Ochoa, J., Pyrcz, M.J., 2022. Fair train-test split in machine learning: Mitigating spatial autocorrelation for improved prediction accuracy. Journal of Petroleum Science and Engineering 209, 109885. https://doi.org/10.1016/j.petrol.2021.109885

Schneider, R., Vicedo-Cabrera, A.M., Sera, F., Masselot, P., Stafoggia, M., de Hoogh, K., Kloog, I., Reis, S., Vieno, M., Gasparrini, A., 2020. A Satellite-Based Spatio-Temporal Machine Learning Model to Reconstruct Daily PM2.5 Concentrations across Great Britain. Remote Sensing 12, 3803. https://doi.org/10.3390/rs12223803

Schratz, P., Muenchow, J., Iturritxa, E., Richter, J., Brenning, A., 2019. Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data. Ecological Modelling 406, 109–120. https://doi.org/10.1016/j.ecolmodel.2019.06.002

Schubert, E., 2023. Stop using the elbow criterion for k-means and how to choose the number of clusters instead. SIGKDD Explor. Newsl. 25, 36–42. https://doi.org/10.1145/3606274.3606278

Statistical Learning: 5.2 K-fold Cross Validation, 2022.

Stojanova, D., Ceci, M., Appice, A., Malerba, D., Džeroski, S., 2011. Global and Local Spatial Autocorrelation in Predictive Clustering Trees, in: Elomaa, T., Hollmén, J., Mannila, H. (Eds.), Discovery Science, Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 307–322. https://doi.org/10.1007/978-3-642-24477-3_25

Tibshirani, R., Walther, G., Hastie, T., 2001. Estimating the Number of Clusters in a Data Set Via the Gap Statistic. Journal of the Royal Statistical Society Series B: Statistical Methodology 63, 411–423. https://doi.org/10.1111/1467-9868.00293

Universidade do Porto, 2024. California housing [WWW Document]. California housing. URL https://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.html (accessed 6.11.24).

van Rossum, G., 1995. Python tutorial. Centrum Wiskunde & Informatica, Netherlands.

Walsh, E.S., Kreakie, B.J., Cantwell, M.G., Nacci, D., 2017. A Random Forest approach to predict the spatial distribution of sediment pollution in an estuarine system. PLOS ONE 12, e0179473. https://doi.org/10.1371/journal.pone.0179473

Wang, Y., Khodadadzadeh, M., Zurita-Milla, R., 2023. Spatial+: A new cross-validation method to evaluate geospatial machine learning models. International Journal of Applied Earth Observation and Geoinformation 121, 103364. https://doi.org/10.1016/j.jag.2023.103364

Wang, Y., Wu, G., Deng, L., Tang, Z., Wang, K., Sun, W., Shangguan, Z., 2017. Prediction of aboveground grassland biomass on the Loess Plateau, China, using a random forest algorithm. Sci Rep 7, 6940. https://doi.org/10.1038/s41598-017-07197-6

Waskom, M., 2021. seaborn: statistical data visualization. JOSS 6, 3021. https://doi.org/10.21105/joss.03021

Web Mercator projection, 2024. . Wikipedia.

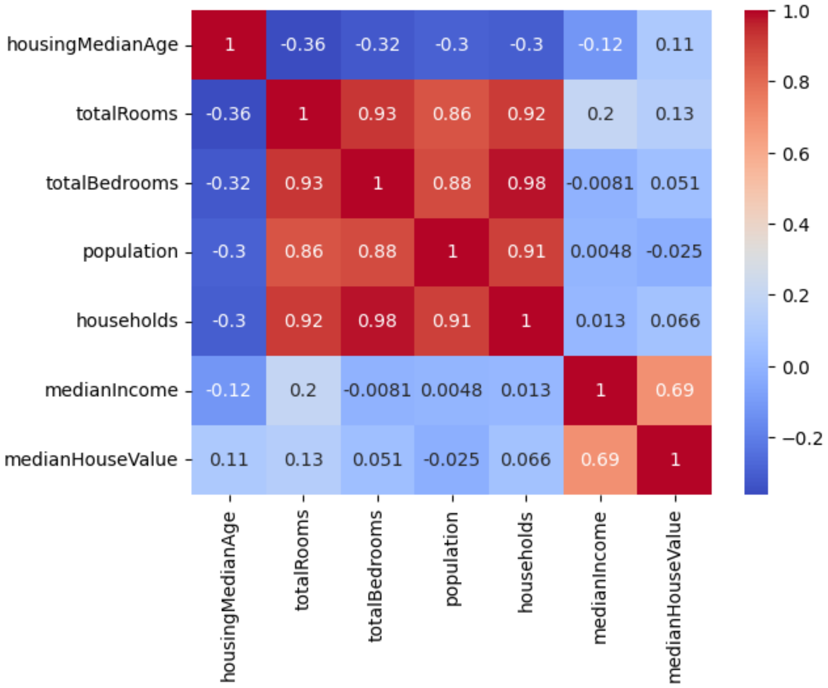
Webster, R., Oliver, M.A., 2007. Geostatistics for Environmental Scientists, 1st ed. Wiley. https://doi.org/10.1002/9780470517277

Weighted arithmetic mean, 2024. . Wikipedia.

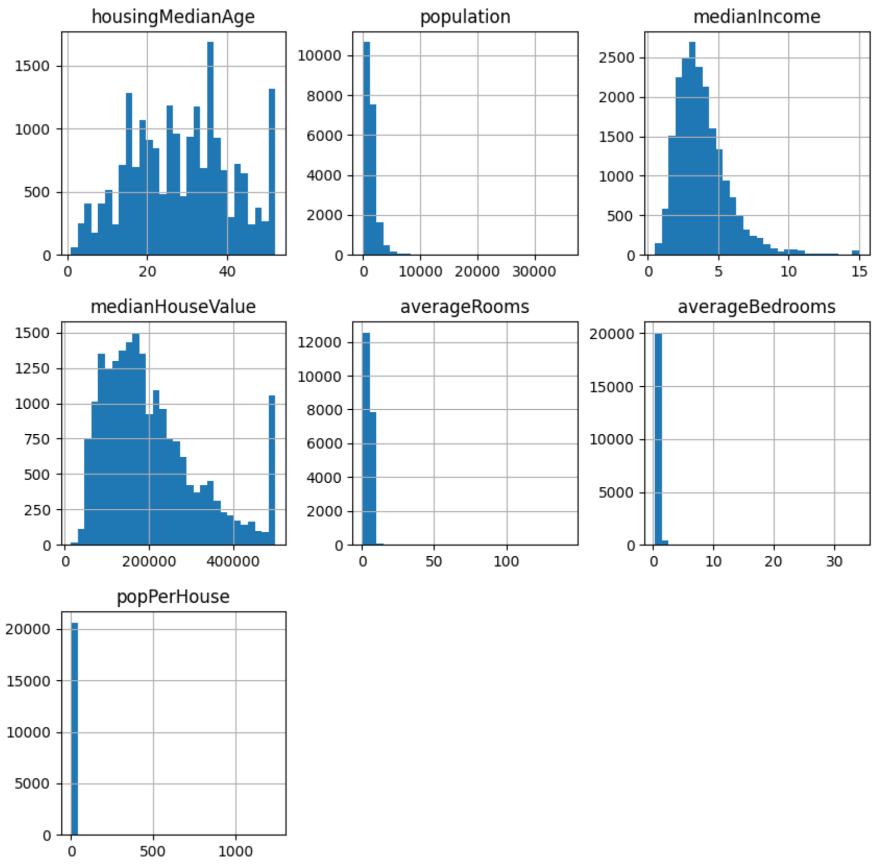
Wenger, S.J., Olden, J.D., 2012. Assessing transferability of ecological models: an underappreciated aspect of statistical validation. Methods in Ecology and Evolution 3, 260–267. https://doi.org/10.1111/j.2041-210X.2011.00170.x

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L.D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T.L., Miller, E., Bache, S.M., Müller, K., Ooms, J., Robinson, D., Seidel, D.P., Spinu, V., Takahashi, K., Vaughan, D., Wilke, C., Woo, K., Yutani, H., 2019. Welcome to the Tidyverse. Journal of Open Source Software 4, 1686. https://doi.org/10.21105/joss.01686

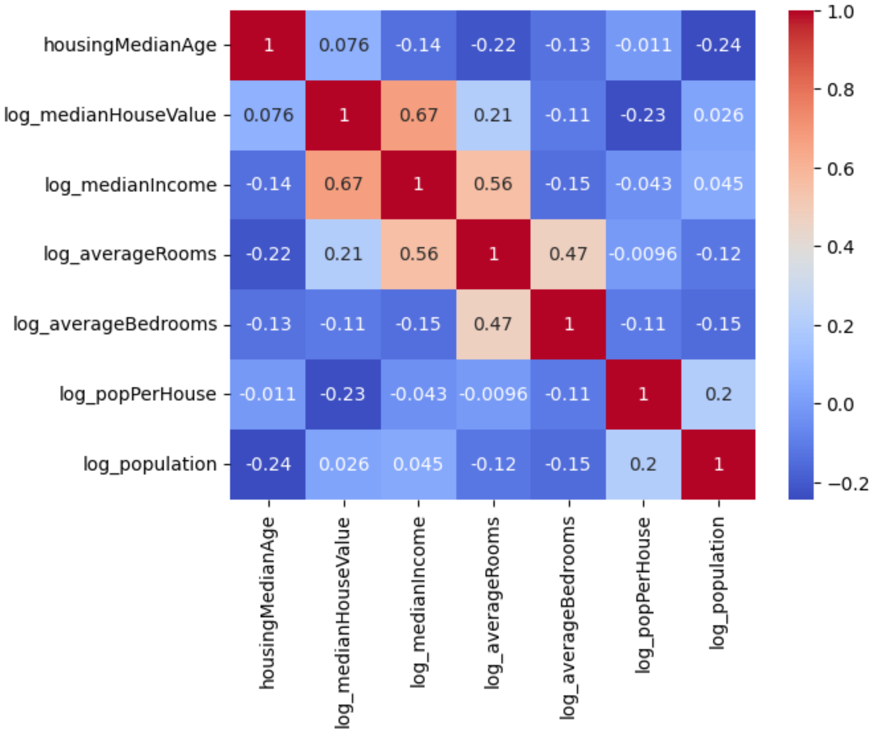
11. Appendix 1.



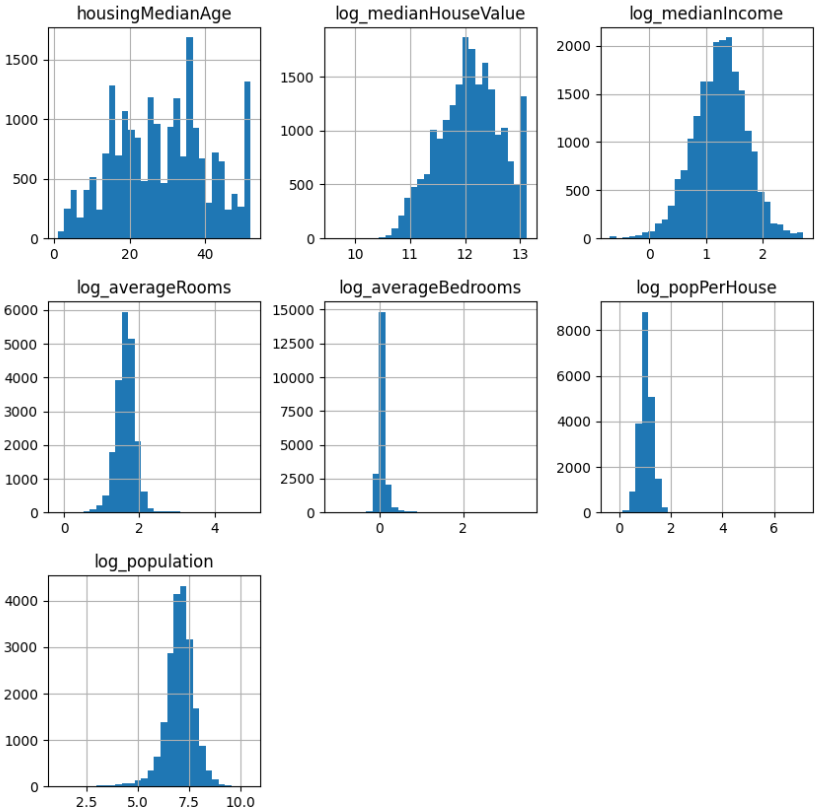
Appendix 1 Figure 1. Correlations between set predictors before feature engineering and log-transformation



Appendix 1 Figure 2. Distributions of predictors after feature engineering but before log-transformation



Appendix 1 Figure 3. Correlations between predictors after feature engineering and log-transformation.



Appendix 1 Figure 4. Distributions of predictors after feature engineering and log-transformation

Appendix 1. Table 1. P-values from the Shapiro-Wilks normality test for validation and test RMSE values for all models. Values in yellow indicate evidence that the distribution of values is significantly different (p < 0.05) from the normal distribution.

Model	Validation RMSE	Test RMSE	Test-valid RMSE difference
random_lag20_predictors	0.141	0.061	0.022
random_lag20_response	0.527	0.415	0.752
random_lag5_predictors	0.591	0.098	0.076
random_lag5_response	0.091	0.254	0.650
random_no_lag	0.644	0.064	0.085
spatial_buff_lag20_predictors	0.018	0.289	0.449
spatial_buff_lag20_response	0.116	0.039	0.374
spatial_buff_lag5_predictors	0.144	0.106	0.326
spatial_buff_lag5_response	0.444	0.254	0.519
spatial_lag20_predictors	0.023	0.557	0.807
spatial_lag20_response	0.195	0.030	0.204
spatial_lag5_predictors	0.178	0.103	0.332
spatial_lag5_response	0.519	0.254	0.482
spatial_no_lag	0.078	0.053	0.247