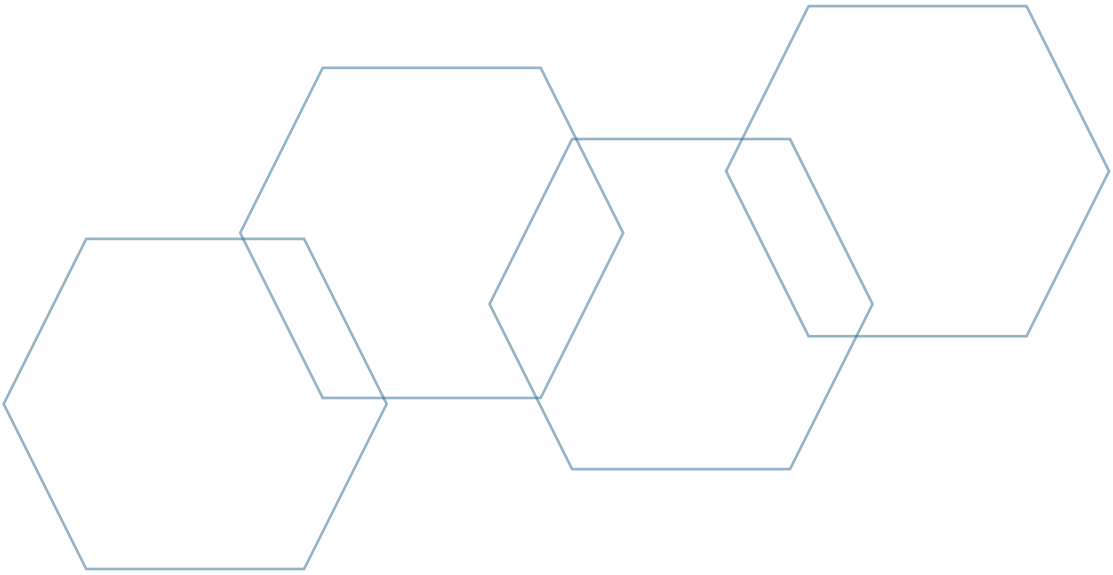
# Can't See the Forest for the Trees

## MSc Thesis GIMA

Flinn Berks

Applying Geospatial Statistics to assess Tree based Machine Learning AVMs in the Dutch Residential Real Estate Market

-



**First Supervisor:**

Abdullah Kara

**Responsible Professor:**

Peter J.M. van Oosterom

03-03-2023

TU Delft

## Preface

I would like to express my sincere gratitude to all those who have contributed to this final MSc thesis project. First and foremost, I would like to thank my supervisor, Abdullah Kara, for his guidance and support throughout the project. I am grateful for his patience and willingness to provide valuable feedback and ideas. I would also like to thank my colleagues at Capital Value, for their help in the development of this project. Their expertise and knowledge have been invaluable in helping me carry this thesis to the final stage. Finally, I would like to thank my family and friends for their encouragement and support throughout the process. Special thanks to my brother Scout for his helpful insights and support. Hopefully, everybody reading this thesis enjoys reading it as much as I enjoyed doing it.

## Abstract

Machine Learning techniques have quickly found their usefulness in the housing market. Valuation and large scale prediction of housing prices are among the most common application. While many different metrics are often used to compare the performance of these techniques to each other, studies neglect how they perform with regards to different variables in a model. Mapping the impact of features in a model in a systematic way allows us to gain more understanding in the choices made by an algorithm. This study researches the impact of different variables on the base configuration of Decision Trees, Random Forest and Gradient Boosting algorithms to find out where prediction errors occur most. The dataset comprised of residential sales transactions in The Netherlands in 2021. The study has found that the different machine learning models show a geographical variance in feature importance on a national scale. This thesis has also shown the importance of several important features that impact the prediction error most, including usable floor area, the number of transactions in an area and the fact that a house is free standing.

# Table of Contents

# 1

# Introduction

## 1.1 The problem and its context

From a purely economic standpoint, a housing market can be seen as a place where buyers and sellers negotiate an agreement on the to-date value of a residence, individually or through a real estate broker. The initial asking price is most often decided by an external appraiser or a real estate broker with market knowledge of a specific geographic region. While appraisers are under strict supervision from regulators, relying on one valuation method for a valuation can cause sensitivity issues. The 2008 housing crisis is an example of what can happen when supervision is insufficiently executed. To counter these issues, one such controlling institution, the Waarderingskamer, stated that machine learning models can be used as a control model for tax-based valuations of properties (Waarderingskamer, 2022). The models that are referred to here are oftentimes called automated valuation models (AVM). These AVMs make use of a form of statistical inference to predict property values according to a set of parameters.

Studies have shown that the results predicted by Random Forest machine learning regression models (RF-MLR) show a deviation from the actual value (Bensdorp, 2021). Additionally, the outcomes of a comparative analysis thesis using Support-Vector regression to estimate housing prices show a significant misprediction in the most expensive neighbourhoods in the Netherlands (Kars, 2021). The results from these two theses, seem to show that there are in fact patterns visible in the geographical accuracy of the prediction outcomes. However, these assumptions are based on visual interpretation. To prove these assumptions this study will try to find patterns and scientifically argument them.

As can be observed in previous studies, the outcomes of Machine learning AVMs are rarely plotted on a map. Most studies don't dive much deeper into what geographical differences can be perceived in such maps. This research poses to fill this knowledge gap by applying geospatial statistics to better understand spatial differences in the outcomes of open source algorithms.

The research for this thesis will be carried out in cooperation with Capital Value. This Utrecht based company is a real estate advisor and appraisal specialist focussing on the Dutch residential real estate market. The assistance from Capital Value in this thesis opens a unique opportunity to take the next step in the scientific argumentation for machine learning AVMs. Additionally, this creates a chance to research market value as opposed to the tax-based value (WOZ) in most literature, which is calculated once a year.

The research consists of comparing the outcomes of multiple machine learning algorithms that predict housing prices in the Netherlands. Furthermore, the research dives into the analysis of possible patterns in geographical prediction accuracy and feature performance of the prediction of residential real estate. In overall conclusion, the research poses to answer the question of what variables play a significant role in geographical accuracy on a national scale.

The scope of this study comprises solely sales transactions and excludes all rent transactions. Transactions in the Netherlands in recent years have shown how much of a role location plays in the eventual value of a real estate property. Additionally, Brainbay, an NVM subsidiary, has shown in research that energy label plays a different role in different regions in the Netherlands (NVM, 2022). These factors are only two of the many that play a role in the valuation of a property. To find out the specific effect of different variables in their geographic region this study will make use of hex maps instead of using conventional administrative units to better analyses emerging spatial dependency. A hexagonal grid poses some aggregation problems; however, literature shows that it is a good method for finding spatial patterns.

## 1.2 Research Questions

The general aim of this research is to find out if there exist geographical patterns in the prediction accuracy for mass appraisal techniques. The research proposes to answer the following main question:

---

*How can geospatial statistics help us better understand local differences in misprediction in state-of-the-art Machine Learning AVMs?*

---

**Sub-questions**

The research poses to answer this question with the use of four sub-questions:

**Q1**. How can we measure the accuracy of individual AVM predictions?

- This part of the research will focus on assessing various metrics to measure the difference in prediction and actual values. The models as well as the individual variables will be assessed on their prediction accuracy. Multiple methods will be discussed in the theory and methods.

**Q2**. What is the spatial autocorrelation of the difference between predicted and actual values?

- Spatial autocorrelation can show the presence of systematic spatial variation in a mapped variable (Haining, 2001). Using spatial variation can improve the dataset as well as visualize complex spatial patterns.

**Q3**. How do different ML AVMs compare in terms of spatial patterns?

- For optimal research, it is necessary to take into account a multitude of algorithms to be able to generalize the research. This study looks at the Random Forest (RF), Gradient Boosting (GB) and Decision Tree Regression (DTR) algorithms for house price prediction.

**Q4**. Which variables play a significant role in places with exceptional hot or cold spots regarding the difference in predicted and actual values?

- This research question will be answered by using spatial regression techniques. The research will try to investigate to what extent the used variables can explain the patterns in misprediction.

## 1.3 Limitations and scope

Within the domain of geographical accuracy in mass appraisal models many interesting topics emerge. During orientation, many gaps in scientific literature occur. Due to the limited time and resources, decisions need to be made regarding the scope of the research. This section will briefly mention interesting directions for future research and why they will be excluded from this thesis.

An important note to consider is the fact that this research will not be ranking current state-of-the-art methods for mass appraisal techniques. However useful for the focus of development on AVMs, this could only be done with more time and a more comprehensive analysis of the models themselves. Additionally, this research will not attempt to create a competitive appraisal model. For this research, currently available modules of existing algorithms will be used. The scope of this research will not go beyond the geostatistical analysis of the outcomes. Temporal influences, like assessing the impact of irregularities in the market (e.g., the financial crisis of 2008) on prediction accuracy are interesting topics to research in the future but will also not be included in this research. However, the dataset used for this study only covers transactions in 2021.

# 2

# Theoretical Framework

## 2.1 The use of machine learning in valuation

Research has found that linear approaches are not the most effective in accurately modelling real estate prices(Ho et al., 2021; Hoang & Wiegratz, 2022; Peterson & Flanagan, 2009). As a result, researchers have turned to methods that are better suited for identifying complex patterns. In recent years, there has been significant research and development on using machine learning techniques in property valuation. This theoretical framework aims to provide an overview of the scientific literature on this topic and to highlight the key principles that underpin machine learning approaches to property valuation. The selection, validation, and interpretation of models are among the most frequently discussed principles in the literature. Machine learning automated valuation models use a data-driven approach to a relatively subjective method, which is traditional appraisal ((Chou et al., 2022)). As property valuation is a data-intensive process that relies on a wide range of information to generate accurate estimates, machine learning algorithms use large datasets to build models that can predict property values accurately. These algorithms are capable of learning from historical data to identify complex patterns and relationships between various factors (Chou & Bui, 2014; Ngiam & Khor, 2019)

Machine learning models have found their way into practice in various academic and commercial fields. According to a meta-analysis study by Chaphalkar & Sandbhor (2013) the academic field has been studying machine learning models in the prediction of housing prices from as early as 1990. Studies on the prediction of housing prices using ML models are often done as case studies on a particular country or region within a country and mostly aim to compare different algorithms. Many geographies have been subject to similar research like the ones on Melbourne (Phan, 2018), London (Ng & Deisenroth, 2015) and Taipei (Chou et al., 2022). Park & Bae (2015) state that location in these automated valuation practices are incredibly important to the outcome. A model trained for a specific region can therefore impossibly operate on different regions. They state that different geographic regions might require different attributes or features. In other words, features don't have the same importance from one region to another.

## 2.2 Fiscal vs market value appraisal

Research has found that generally there is a significant difference in the market value as opposed to the fiscal value of a property (Lubberink et al., 2017). While the land administration domain model (LADM) is a knowledge domain specific standard capturing the semantics of the land administration domain, it does include many intrinsic characteristics of a building that are important to predict the value of a house. The initial model strains the importance for documenting the relation between people and land (Lemmen et al., 2022). The LADM-based valuation information extension has been developed for the specification of valuation information maintained by public authorities. The proposed model was first introduced in Cagdas et al. (2016) and has been reviewed, revised and improved, taking into account the comments received from the various workshops, including the 7th, 8th and 9th FIG LADM workshops. The latest version of LADM_VM is designed to facilitate all stages of administrative property valuation, namely, identification of valuation units, valuation of units through individual or mass valuation procedures, recording of transaction prices, presentation of sales statistics and handling of appeals. The extension considers more intrinsic features of a property, including the building type, number of dwellings and the date of construction.

While the root of this extension is designed for fiscal purposes, it also has its application in market valuation. There are some important differences between the two types of valuation to take into account. The first of which is that a fiscal valuation is always determined on a specific time. This therefore does not take into consideration the current state of the house or market situations. Where tax based valuation is done by estimating a price based on objective properties, market valuation may consider many more factors, like the quality of the inside of a house or the market sentiment and emotion.

## 2.3 Model selection

There is a wide range of machine learning algorithms that can be used for property valuation. Some well-known algorithms include decision trees, Random Forests, support vector machine learning (SVM) and artificial neural networks (ANN). The choice of algorithm will depend on the specifics of the problem, such as the size of the dataset and the complexity of the relationships between variables. Ho et al. (2021) conclude that the Random Forest method and the Gradient Boosting method are able to generate accurate price estimations and have lower prediction errors compared to the SVM method. Tree models differ from linear models in the fact that they are able to determine non-linear relations (Guliker et al., 2022). While, several methods of forest based algorithms have been researched, the simplest of all, the singular decision tree has been mostly neglected. This study will also include this version of machine learning and test its relevance in accordance to other models. One study shows similar behaviours in feature importance between the decision tree and Random Forest methods (Beimer & Francke, 2019).

## 2.4 Model validation

Machine learning models must be validated to ensure that they are reliable and accurate. This typically involves splitting the dataset into a training set and a test set, and evaluating the model's performance on the test set. The objective is to avoid overfitting, which is when the model is too closely tailored to the training data and performs poorly on new data. Frequently used methods for assessing the model include the mean squared error (MSE), root mean squared error (RMSE) and mean absolute percentage error (MAPE) (Beimer & Francke, 2019; Ho et al., 2021). Other studies also include the use of the median absolute percentage error (MDAPE)(Aladag et al., 2012) . In addition to these measures, the $R^2$ can be used to measure the explained variance in the model. With regards to spatial statistics, the Moran's I will be added to this list to measure whether the level of misprediction for each transaction is spatially clustered. This additional measure gives us an insight into how these models operate geographically on a national scale. A study on house sale price prediction in Fairfax County, Virginia has taken into consideration the Moran's I (Hu et al., 2022). The authors have found that this metric can be used to correct over and underpredicted house sale prices.

## 2.5 Model interpretation

"*Understanding why a model makes a certain prediction can be as crucial as the prediction's accuracy in many applications*." (Lundberg & Lee, 2017, p. 1). Because they are not well understood, it seems counterintuitive to state that machine learning models are more transparent than traditional methods. However, with the correct metrics, machine learning models can show us a lot about how the market has operated in the past using concepts as feature importance and Shapley values.

The ability to interpret machine learning models is crucial for understanding their predictions and for improving the accuracy of the models. One approach to model interpretation is to use feature importance, this is a broad term for many metrics to measure the relative importance of each variable in the model (Breiman, 2001). Another approach is to use partial dependence plots, which plot the effect of a single variable on the model's prediction while holding all other variables constant. Another metric for determining the feature importance is the impurity-based feature importance . These measures of feature importance are often used for decision trees. The higher, the more important the feature. The importance of a feature is computed as the (normalized) total reduction of the criterion brought by that feature (Scikit Learn, 2023). It is also known as the Gini importance. Another method for determining the feature importance is the feature permutation importance. This method is more useful as it shows less bias towards variables with high cardinality.

The Shapley value is a way to allocate the total prediction value (or the overall contribution) among the features in a model (Lundberg & Lee, 2017). It is based on the idea that each feature should receive credit for its contribution to the prediction, taking into account all possible combinations of features. The Shapley value is calculated by averaging the marginal contribution of each feature over all possible combinations of features, and it has several desirable properties, such as being linear, symmetric, and efficiently computable.

In machine learning, Shapley values have been used to explain the contribution of each feature to the prediction of a model, as well as to identify the most important features in a model and to evaluate the performance of different models (Frye et al., 2020). Shapley values can be applied to various types of models, including linear models, decision trees, and neural networks.

Interpreting the Shapley values requires some understanding of the model and the features used in the model however, not as much as in a linear regression. In general, the Shapley values provide insight into the feature importance and can be used for various purposes, such as:

Feature importance: The Shapley values can be used to rank the features in terms of their contribution to the prediction. Features with higher Shapley values are considered to be more important in terms of explaining the prediction.

Model Explanation: The Shapley values can be used to explain the predictions of a model. By identifying the features that have a positive or negative contribution to the prediction, it is possible to understand how the model makes its predictions.

Model Comparison: The Shapley values can be used to compare the performance of different models by comparing the contributions of the features. For example, it is possible to compare the contributions of the features in two or more different models to understand the similarities and differences between the models.

In summary, the Shapley values provide a valuable tool for understanding the behaviour of a model and for interpreting its predictions. By considering the contribution of each feature, it is possible to gain insight into the model's behaviour and to make decisions about model selection and feature engineering. Shapley values and feature importance are already widely used to assess the performance of a machine learning model. However, these measures have not yet been assessed on their spatial variance.

## 2.6 Take away from literature

Ho et al. (2021) state that in order to improve the application of machine learning in property valuation, data of a larger geographical region is necessary. To understand patterns in feature importance, this study therefore focuses on the whole of the Netherlands. The availability of data points plays a key role in research surrounding data driven methods. While assessing fiscal models may have more direct societal relevance, the availability of transaction data dictates that the research will pose to assess feature importance of market transactions. This research adds to current literature in the fact that it considers the geographical distribution of feature importance.

# 3

# Methodology

## 3.1 General Overview

This research utilizes a five-stage approach (Figure 1) to analyse the dataset. The first stage involves data handling, which includes preparing the dataset for further use. This includes transforming and cleaning the data, as well as selecting features from the dataset. The second stage is the development of three machine learning algorithms, which will be used to generate an output. The third stage involves structuring the output so that it can be used in further analysis. Finally, maps will be created to illustrate the results and help better understand the conclusions.

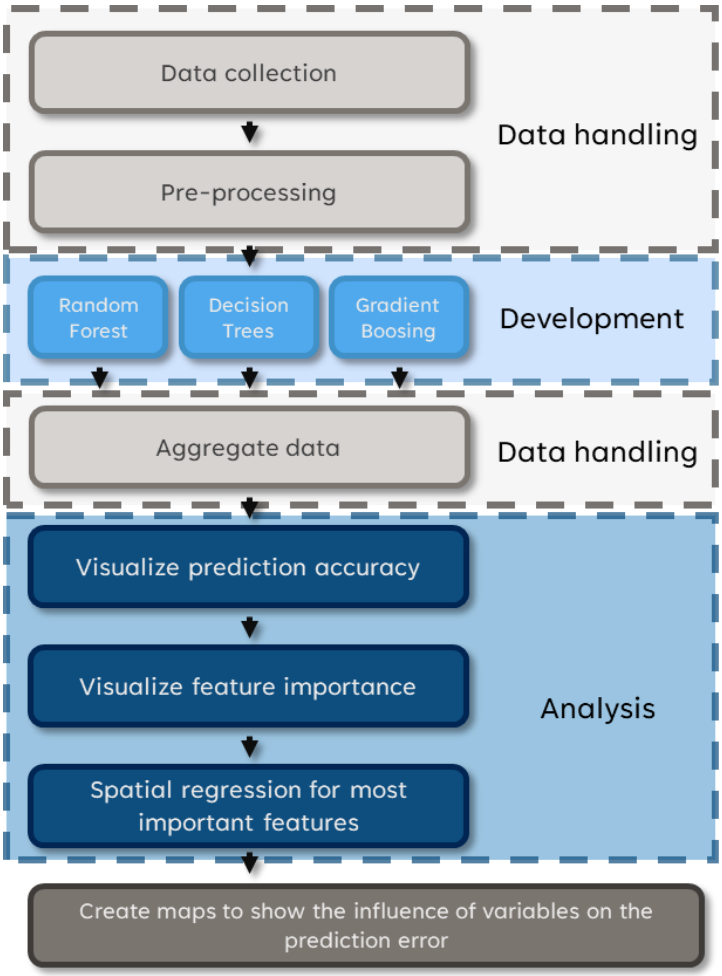



*Figure 1 general overview of methods*

This section of the thesis presents insights into the tools and methodology used in the research. The different stages and the methods used in each stage are discussed in detail in the following sections. Sections 3.1 and 3.2 discuss the data collection and data handling respectively. Section 3.3 then explains the development of the three algorithms. Section 3.4 shows the steps and considerations preceding the analysis, while Section 3.5 the analytical methods.

## 3.2 Data Collection

The data necessary to achieve the projected results consist of real estate transactions for single-family and multi-family homes in the Netherlands from the year 2021. Literature speaks of three main variable groups that are used to predict values. These three groups are structural variables, locational variables and neighbourhood variables (Helbich et al., 2013). In addition to these three groups, this study makes two

additional groups of variables, spatial variables, and market sentiment. The data about structural variables was made available for use by Capital Value. The dataset used is received from Brainbay which operates and maintains the data for the NVM, which is the Dutch Association for Real Estate agents and appraisers. The algorithms were trained with 38 features which includes the independent variable (table 1).

The first dataset used for this study is made available by Capital Value and contains information on the structure and transaction value for the instances. Furthermore, this data contains essential information on the location of the transactions This dataset consisted of 400,000 instances and comprised 55 features per instance (Figure 2a). It is important to note that not all features can be used because this information has not been supplied by the broker at the time of the transaction and is therefore seen as empty values.

The data is then further enriched with locational and neighbourhood variables by joining the '*kerncijfers wijken en buurten 2021'* dataset from CBS. The CBS dataset was first cleaned and rid of unnecessary information (Figure 2b). The CBS dataset consisted of shapefile features in the same coordinate system which made it easy to join the two data sources using a spatial join.

The relationships between all the variables for use in a geodatabase are presented in Figure 3. Because this study is limited to a dataset covering transactions in 2021 only, building a database is not necessary. However for future research, it might be necessary to incorporate the use of a geodatabase as the number of datapoints will grow significantly.

*Table 1 List of Variables (* is a dummy variable)*

| # | Feature name | data type | Variable type | description | Source |
|---|---|---|---|---|---|
| 0 | OrigID | Integer | | Original ID | - |
| | | | **Spatial variables** | | |
| 1 | RD_X | float | Spatial variable | X coordinate RD New | Brainbay, 2022 |
| 2 | RD_Y | float | Spatial variable | Y coordinate RD New | Brainbay, 2022 |
| 4 | GRID_ID | Integer | Spatial variable | The hexagon ID to which the transaction is aggregated | Own calculation |
| | | | **Independent variable** | | |
| 5 | Transactionprice_m2 | float | Ratio | Transaction price per m² | Brainbay, 2022 |
| | | | **Market sentiment variables** | | |
| 6 | Spread_% | float | Ratio | Difference between asking price and transaction price (transaction price per m² - asking price per m² as a percentage of the transaction price per m²) | Own calculation |
| 7 | Days_on_market | Integer | Ratio | The days between offering and the transaction | Brainbay, 2022 |
| 8 | SalesMonth | Integer | Nominal | The month in which a sale has taken place | Brainbay, 2022 |
| | | | **Structural variables** | | |
| 9 | Floor_Area | Integer | ratio | Usable floor area in m² | Brainbay, 2022 |
| 10 | Number_of_rooms | Integer | ratio | The number of heated rooms in a residence | Brainbay, 2022 |
| 11 | BuildingType | object | Ordinal* | Type of residence | Brainbay, 2022 |
| 12 | Energylabel | object | Ordinal* | Energy label A-G (A+++++ - A+ have been put into the A bucket) | Brainbay, 2022 |
| 13 | YearBuilt | Integer | interval | The year in which a residence was built | Brainbay, 2022 |
| 14 | Quality_inside | Integer | ordinal | Quality inside (Bad - Excelent) | Brainbay, 2022 |
| 15 | Quality_outside | Integer | ordinal | Quality outside (Bad - Excelent) | Brainbay, 2022 |
| 16 | Status | object | Nominal* | Newbuilt or existing stock | Brainbay, 2022 |
| | | | **Locational variables** | | |
| 17 | Dist_supermarket | float | ratio | The average distance of all residents in a neighbourhood to the closest supermarket | CBS, 2021 |
| 18 | Dist_daily | float | ratio | The average distance of all residents in a neighbourhood to the closest daily provision stores | CBS, 2021 |
| 19 | Dist_restaurant | float | ratio | The average distance of all residents in a neighbourhood to the closest restaurant | CBS, 2021 |
| 20 | Dist_hotel | float | ratio | The average distance of all residents in a neighbourhood to the closest hotel | CBS, 2021 |
| 21 | Dist_primaryed | float | ratio | The average distance of all residents in a neighbourhood to the closest primary school | CBS, 2021 |
| | | | **Neighbourhood variables** | | |
| 22 | PopDensity | Integer | ratio | Population density in a neighbourhood | CBS, 2021 |
| 23 | P_00_14_AGE | Integer | ratio | Percentage of people aged 0 - 14 in a neighbourhood | CBS, 2021 |
| 24 | P_15_24_AGE | Integer | ratio | Percentage of people aged 15 - 24 in a neighbourhood | CBS, 2021 |
| 25 | P_25_44_AGE | Integer | ratio | Percentage of people aged 25 - 44 in a neighbourhood | CBS, 2021 |
| 26 | P_45_64_AGE | Integer | ratio | Percentage of people aged 45 - 64 in a neighbourhood | CBS, 2021 |
| 27 | P_65_EO_AGE | Integer | ratio | Percentage of 65_ person households in a neighbourhood | CBS, 2021 |
| 28 | P_ONEP_HH | Integer | ratio | Percentage of one person households in a neighbourhood | CBS, 2021 |
| 29 | P_WEST_IM | Integer | ratio | Percentage of Western immigrants in a neighbourhood | CBS, 2021 |
| 30 | P_N_W_IM | Integer | ratio | Percentage of Non-Western immigrants in a neighbourhood | CBS, 2021 |
| 31 | WOZ | Integer | ratio | Average tax based values in a neighbourhood | CBS, 2021 |
| 32 | P_Before2000 | Integer | ratio | Percentage of houses built before 2000 in a neighbourhood | CBS, 2021 |
| 33 | P_after2000 | Integer | ratio | Percentage of houses built after 2000 in a neighbourhood | CBS, 2021 |
| 34 | P_Empty | Integer | ratio | Percentage of vacant houses in a neighbourhood | CBS, 2021 |
| 35 | RAD1_SUPERM | float | Ratio | Number of supermarkets in a one kilometre radius | CBS, 2021 |
| 36 | RAD1_Daily | float | Ratio | Number of daily provision stores in a one kilometre radius | CBS, 2021 |
| 37 | RAD1_RESTAU | float | Ratio | Number of restaurants in a one kilometre radius | CBS, 2021 |
| 38 | RAD1_Primaryed | float | Ratio | Number of primary schools in a one kilometre radius | CBS, 2021 |

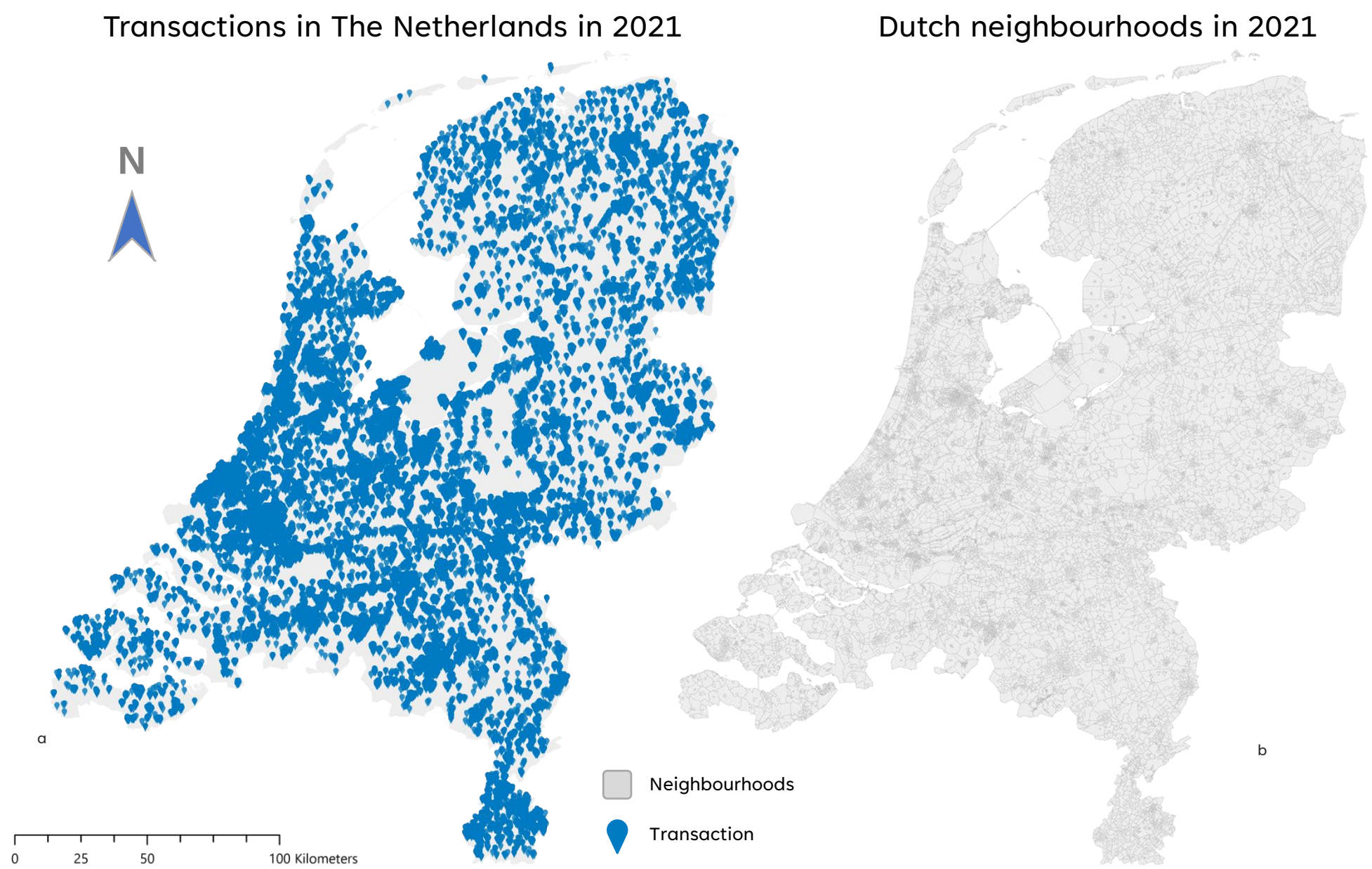



*Figure 2 left: transactions in the dataset; right: administrative borders of neighbourhoods in the Netherlands*

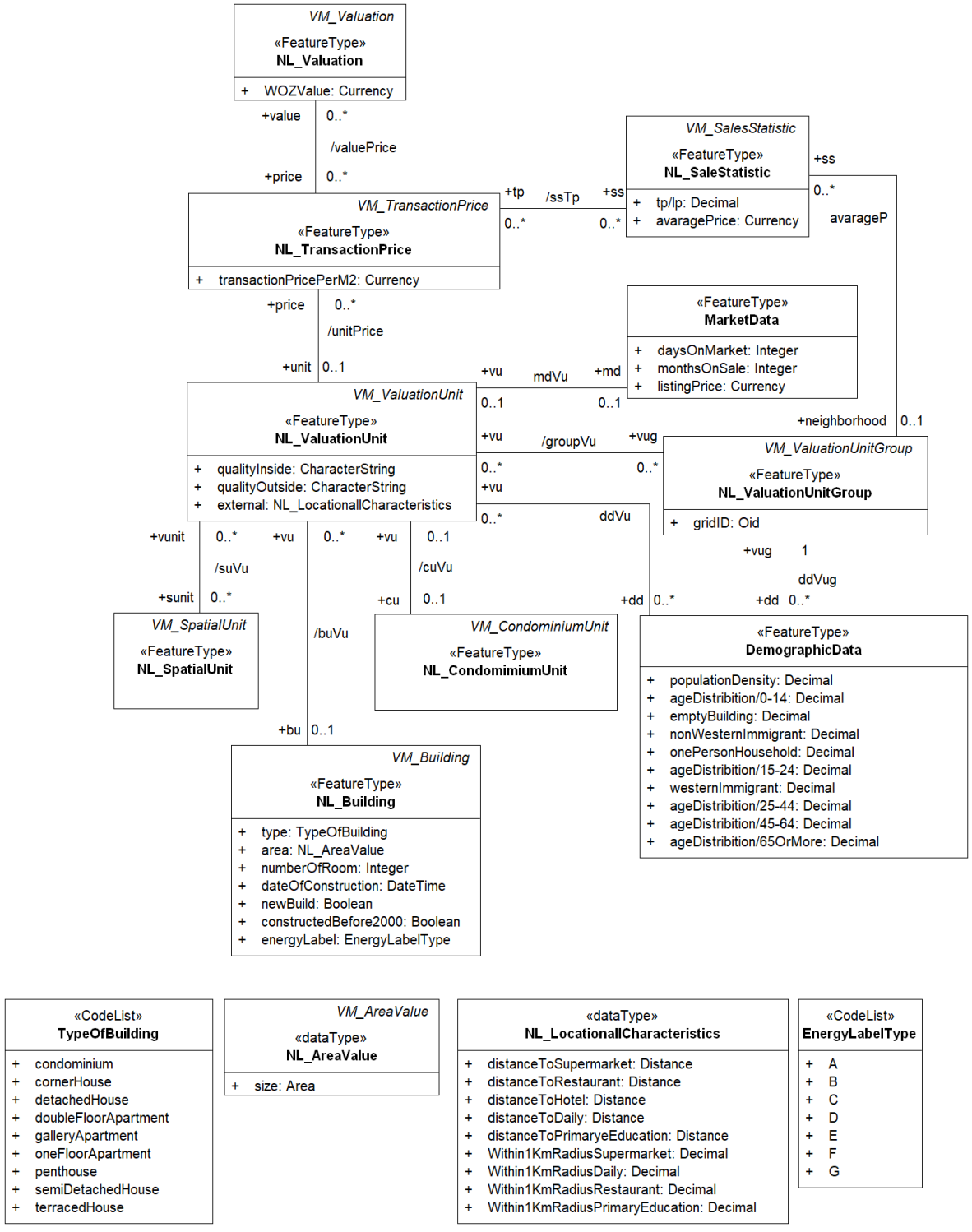



*Figure 3 UML diagram of features used*

## 3.3 Data pre-processing

Pre-processing of data consists of feature extraction and normalization. It is often done to optimize the dataset to ensure significance and a reliable outcome of the models' outputs. The International Association of Assessing Officers (IAAO) developed standards for the use of transaction information. Firstly, they state that the data used for an AVM should pass the following screening tests (1) The dataset should be big enough to

represent the population. (2) Transactions should be valid and need to reflect market value. (3) Property characteristics should be accurate. (4) Sales data and characteristics should be representative of the population (International Association of Assessing Officers, 2018).

In another standards report, they state that transaction data is more valuable when reviewed (International Association of Assessing Officers, 2018). Polluted datasets are not uncommon in the real estate sector. Krause & Lipscomb (2016) point out that field standardization is a problem when dealing with data from multiple input sources. While the data for this thesis comes from one database, not all fields are perfectly standardized. Additional relevant problems they mention are data errors and missing data. With regards to GIS problems, missing data or data errors in the address section might have implications for geocoding the data.

To rid the dataset of inconsistencies, the IAAO propose methods for identifying outliers by visual examination on a GIS map or by plotting the information on graphs such as a scatter or a box plot (International Association of Assessing Officers, 2016). In accordance with this, Bidanset & Lombard (2014) promote the use of an IQRx3 approach to detect outliers in the dataset. It differs from the regular method in that only very extreme values are removed. (Davis et al., 2020) used Cook's Distance and Mahalanobis Distance to estimate the effect of outliers on the accuracy of the data. Additional methods to detect outliers include the use of global and local Moran's I as used in research on residential exposure to pollution and noise (Verbeek, 2018).

**Building status**

As a first step to cleaning the dataset all data that didn't meet the requirements of being a sales transaction in the year 2021 were removed. This resulted in a little under 200,000 transactions. After this step duplicates and imprecise instances were removed from the dataset. The method for finding duplicates was to create a string from the X and Y coordinate. The coordinates are in accordance with the BAG location and will therefore have limited overlap. However, this poses a significant problem for newly built residences.

Oftentimes these houses have not been built before they are sold, which means that the location for a new-built residence overlaps with the other residences within that project. This method therefore reduces the number of new-built residences significantly to a point where they cannot be accurately predicted anymore (Figure 4). This would not be a problem if the correct BAG-ID is supplied with the data. Unfortunately, this is not the case for the available dataset used in this thesis.

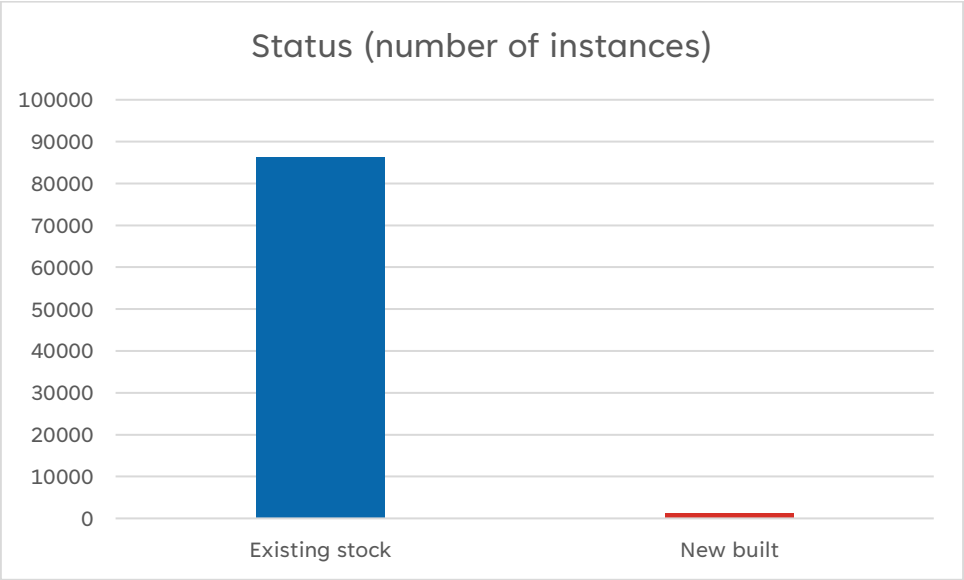



*Figure 4 Number of instances per building status*

**Building types**

When the dataset was rid of most null values the categorical features were analyzed and checked for inconsistencies. Besides building status, the two categorical variables are building type and energy label. When examining building type, the conclusion was drawn that some building types showed overlap in their features. For this reason, new bins were created for residence types with somewhat similar features (Figure 5). The benefit of doing this is that the categories become larger, hopefully resulting in a more significant outcome. Other housing types like farmhouses, country houses and estates were removed from the dataset altogether. When applying a spatial join with the CBS data, it was possible to select the data on the average tax based value of the neighbourhood. Due to the criteria set by the CBS, it was possible to further specify the dataset according to these parameters. For deciding the average tax based value per neighbourhood only houses with a residential primary function and a tax based value between 10 thousand and 5 million euros were considered.

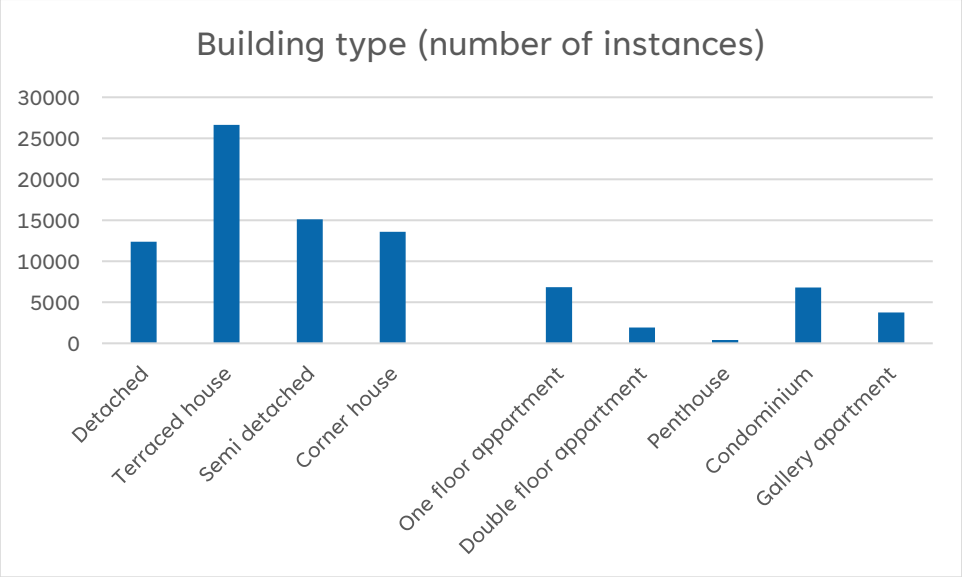



*Figure 5 Number of instances per building type*

**Energy label**

The energy label has proven to be important over the years (NVM, 2022). A study by Brainbay shows that the average added value for going from energy label G to C is about €25,000. They state that this added value has been relatively stable between the period of 2018 and 2021, but that in 2022 this is increasing to up to €35,000. The stability between the period of 2018 and 2021 can be attributed to the housing shortage, leading to a fewer range of choices for homebuyers. In this research, they state that the regional difference in importance is a direct effect of the housing shortage in a specific region. These outcomes are all based on the Brainbay AVM. The division of energy labels is visible in Figure 6. Not all transactions were supplied with an energy label. The choice was made to exclude all transactions without energy labels to make sure the dataset had no missing values. Taking all these measures into account resulted in a dataset of 87,446 instances.

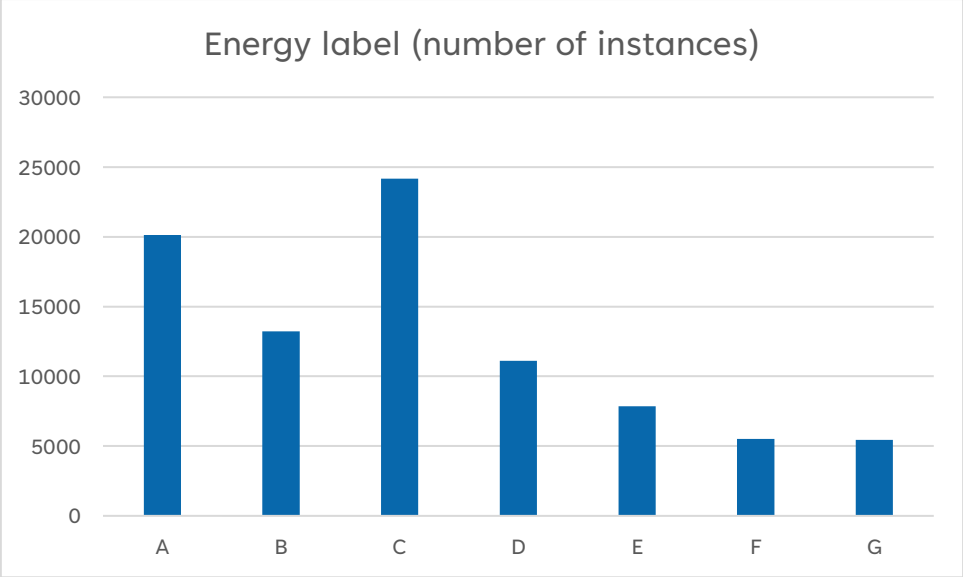



*Figure 6 Number of instances per energy label*

### 3.4.1 Development of the algorithms

The three prediction models make use of three different types of tree-based machine learning algorithms to predict the housing prices. The algorithms are imported from a module developed by SkLearn. The three scripts are similar to each other in their parameters and only differ in which regression algorithm is used. All Decision Tree algorithms are based on the same principles. First of the algorithm selects a random subset of the training data to use as the input for every tree. Secondly, the algorithm creates a decision tree using the subset created in the first step. The tree (or model) grows until a stopping criterium is met. While the algorithms are loosely based on the same principles there are some key differences that set them apart from each other.

**Method**

Random Forest is an ensemble method that uses multiple decision trees to make predictions, whereas Decision Tree is a single tree model. Gradient Boosting is also an ensemble method, but it uses a series of decision trees, where each tree tries to correct the mistakes made by the previous tree in the sequence. Random Forest uses a large number of trees, typically hundreds or thousands, whereas Gradient Boosting uses a smaller number of trees, typically dozens or a few hundred. The Decision tree method is a single tree model as opposed to the other two methods (Mueller & Massaron, 2021).

**Bagging vs Boosting**

Random Forest is based on the bagging technique, which stands for bootstrap aggregating. The idea behind bagging is to reduce the variance in the model by combining the predictions of multiple trees. On the other hand, Gradient Boosting is a boosting technique that focuses on reducing the bias of the model by iterating over a sequence of trees that correct the mistakes made by the previous trees (Plaia et al., 2022).

**Overfitting**

Random Forest is less prone to overfitting compared to Decision Trees, due to its ensemble nature. Gradient Boosting is also less prone to overfitting than Decision Trees, but it can still overfit if not properly tuned (Mueller & Massaron, 2021).

To summarize, each of these algorithms has its own strengths and weaknesses and the choice of which one to use depends on the specific problem and the data at hand. Random Forest is generally considered to be a more robust algorithm for complex and large datasets, while Decision Trees can be easier to interpret and understand, making them a good choice for small and simple datasets. Gradient Boosting is often used when a high level of accuracy is desired, but it can be more difficult to interpret and more time-consuming to train.

### 3.4.2 The script

The three pieces of code were used to create a model and assess its performance. It begins by importing the necessary libraries, such as pandas, numpy, seaborn, matplotlib, and shap. It then loads the prepared datafile from a csv document and creates the X (Independent variables) and y (target variable) variables. Next, the code splits the data into train and test sets using the train_test_split function from sklearn (Scikit learn, 2022). This step requires the definition of the size of the training and test set. For the purposes of this research it is important to acknowledge that the more training samples we provide the algorithm, the less testing samples we get to make maps out of. While the division of the training and test dataset is arbitrary and customized for each purpose, a recognised split is 80-20 (Yilmazer & Kocaman, 2020). However, other commonly used ratios are 70:30, 60:40 or in some cases even 50:50 (Joseph, 2022). In order for the maps to show enough values it was chosen to set the size of the training dataset to 66% of the whole dataset. The same training and test dataset was used for all maps to ensure that the results are comparable.

It then creates a regressor model and fits it to the training data. The model is then used to predict the values of the test data. After this the Tree Explainer object is called which calculates Shapley values. It also calculates the feature importance values from the model and creates a data frame with the feature names and importance values. The code then concatenates the two tables with the predicted and actual values, as well as the Shapley values. It then calculates several statistical scores to assess the performance of the model, such as mean squared error, root mean squared error, mean absolute percentage error, and median absolute percentage error. Finally, the code saves the results, performance scores, and feature importance values to CSV files that can be used in further analysis. To enhance the transparency of the study the scripts are added to the annex (annex A-C).

## 3.5 Visualisation & Data Aggregation

The data used cannot be published on a transaction level due to the sensitivity of this data. For this reason, hexagons will be used to aggregate and therefore anonymise the data. The use of 'hexmaps' or equal area

unit maps (EAUMs) can significantly increase the detection rate of local extreme values (Schiewe, 2021). The administrative boundaries in the Netherlands are not all the same shape and size and are therefore not ideal for aggregation purposes for the goal of analysing patterns.

To increase the visualization of the maps and to be able to perform good analysis this study will therefore use an aggregation method to display the data. To be able to aggregate the transactions to a readable hex map it was first necessary to decide on the cell size. To counter the negative effects of the modifiable areal unit problem (MAUP) it would be considered best practice to show the maps in multiple levels of aggregation. While the author is aware of the fact that the aggregation method can greatly influence the readings of the map, it is decided that the study will only use one measure for the aggregation into hexagons (Chen et al., 2022).

Before cleaning, the CBS dataset consisted of 14,175 administrative units (neighbourhoods). This number still includes all the sections that cover bodies of water, these were later removed to make the average neighbourhood area more precise. The number of neighbourhoods displayed by their area size is displayed in Figure 7. The two ways for deciding the neighbourhoods are taking the mean and taking the median neighbourhood size (Figure 8 a-b). The mean neighbourhood size results in roughly the same number of cells as neighbourhoods while the median neighbourhood size is more compliant with the average size of a neighbourhood. The reason why this is important is because it will decide the number of instances within each cell. The rule of thumb is here that the smaller the surface, the more spatially precise the data. However, if the hexagons are made too small, the surface will only cover a few instances, making the aggregation inadequate. To be able to distinguish patterns in more sparsely populated areas it is decided to use the mean neighbourhood size as the size definition for the hexagonal cells. Another added benefit of using a larger cell size is that more cells will share a common border. This implies that detecting patterns will be easier when adding contiguity weights during the regression. The number of grid cells therefore roughly aligns with the number of neighbourhoods (15,232). Another factor to take into consideration is that the hexagonal grid does not line up exactly with the CBS administrative boundaries. To overcome this, the transaction data will first be enriched with the neighbourhood data before aggregating it to the hexagonal grid.

When aggregating the data another factor needs to be carefully considered, namely the type of aggregation. Of all transaction the mean value was used except for sales month, for this variable the mode was used. For the type of house, the energy label and the status of the building the sum was used. This was considered best as these are ordinal values and are read as dummy variables and don't allow for averaging

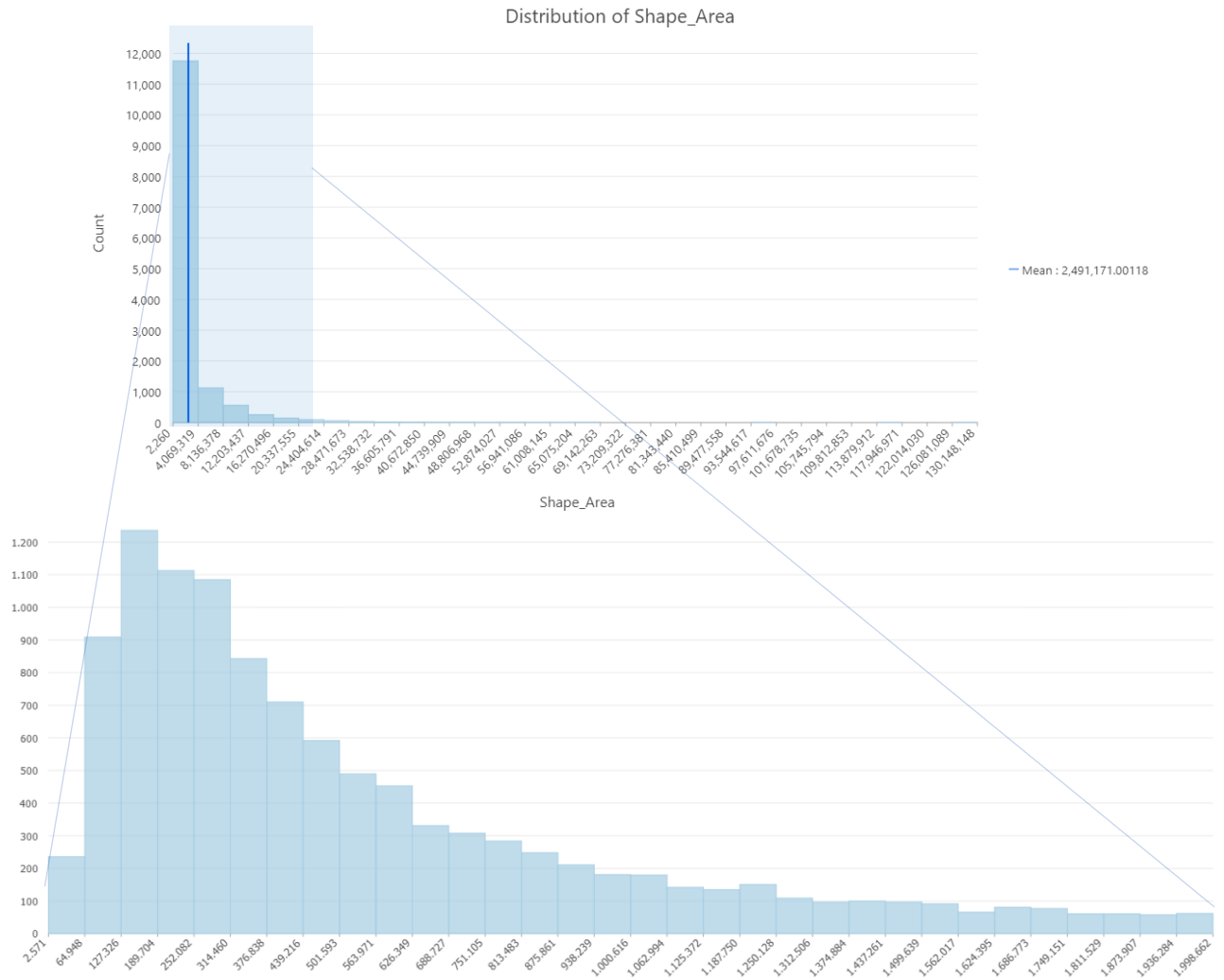



*Figure 7 Distribution of neighbourhood sizes (m²)*

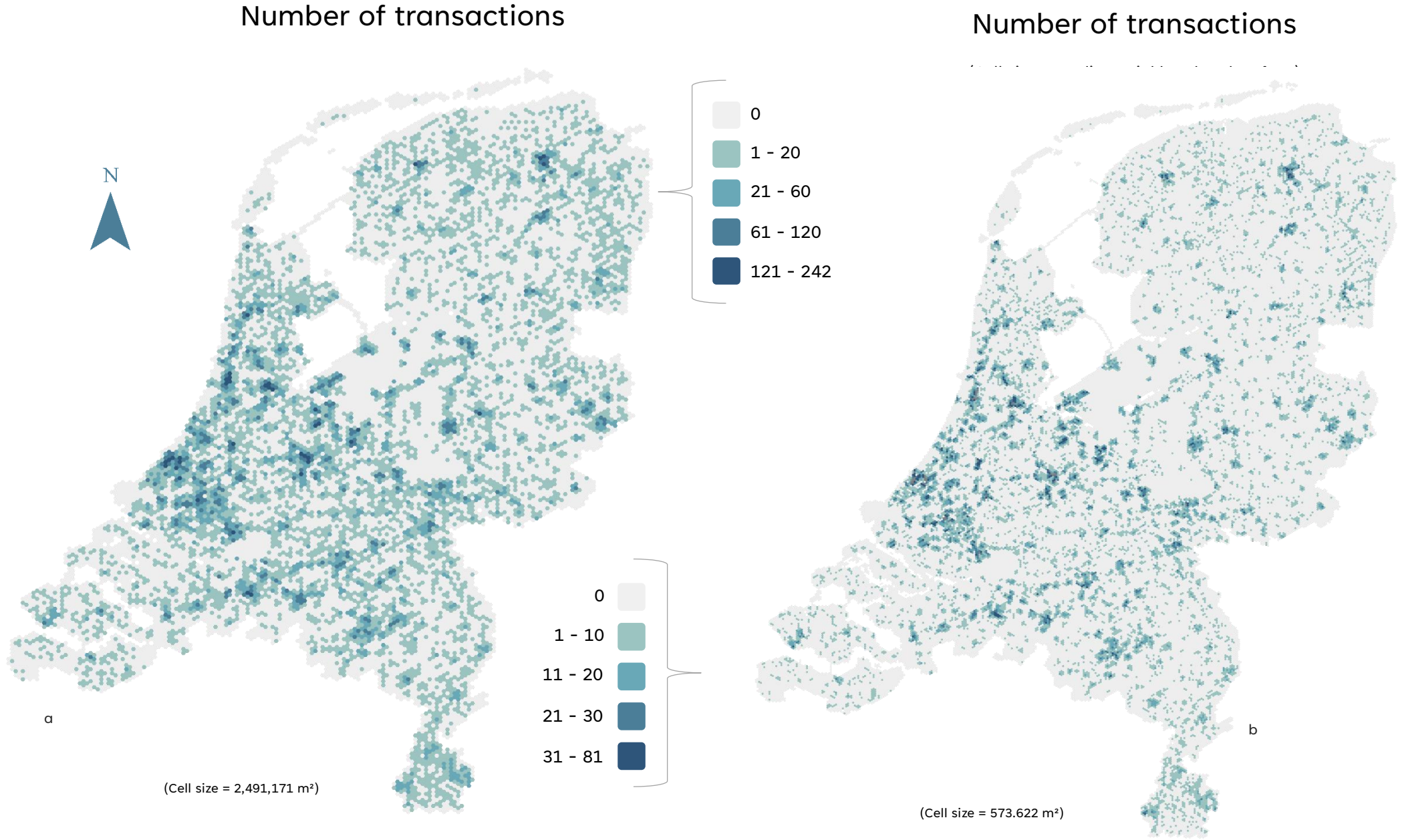



*Figure 8 number of transactions aggregated to different cell sizes*

## 3.6 Exploratory data analysis

With all the data in place and cleaned, it is now possible to perform some simple exploratory data analysis. This section will dive deeper into the spatial distribution of the dataset and will find some preliminary patterns which will be taken into account later in the study. First, the data was explored based on the principal input for this data, the number of transactions (Figure 8). We can see here that most transactions have taken place in and around the Dutch Randstad. Furthermore, we can see that the highest transaction prices have occurred in the G5 (Amsterdam, Rotterdam, The Hague, Utrecht & Eindhoven) (Figure 9 a-b).

To further explore the variables in the dataset a heatmap of the variables was made (Figure 10). This heat map shows the Pearson correlation for all variables. One of the reasons to plot the correlation of the variables in a heat map is to show which dependent variables correlate with each other. This measure is necessary to prevent multicollinearity. Regarding this, we see that the neighbourhood variables that deal with the number of stores in a neighbourhood correlate relatively highly with each other. Meaning that in neighbourhoods with a lot of big supermarkets, there are also a lot of daily provision stores and restaurants. The same is true for the average distance to the closest of these stores. We can also see that the quality of the house on the inside correlates with the quality outside and that they are of importance to the average transaction price per $m^2$. Another interesting albeit somewhat logical interpretation of this heat map is the fact that the number of stores correlates negatively with the average distance to those stores.

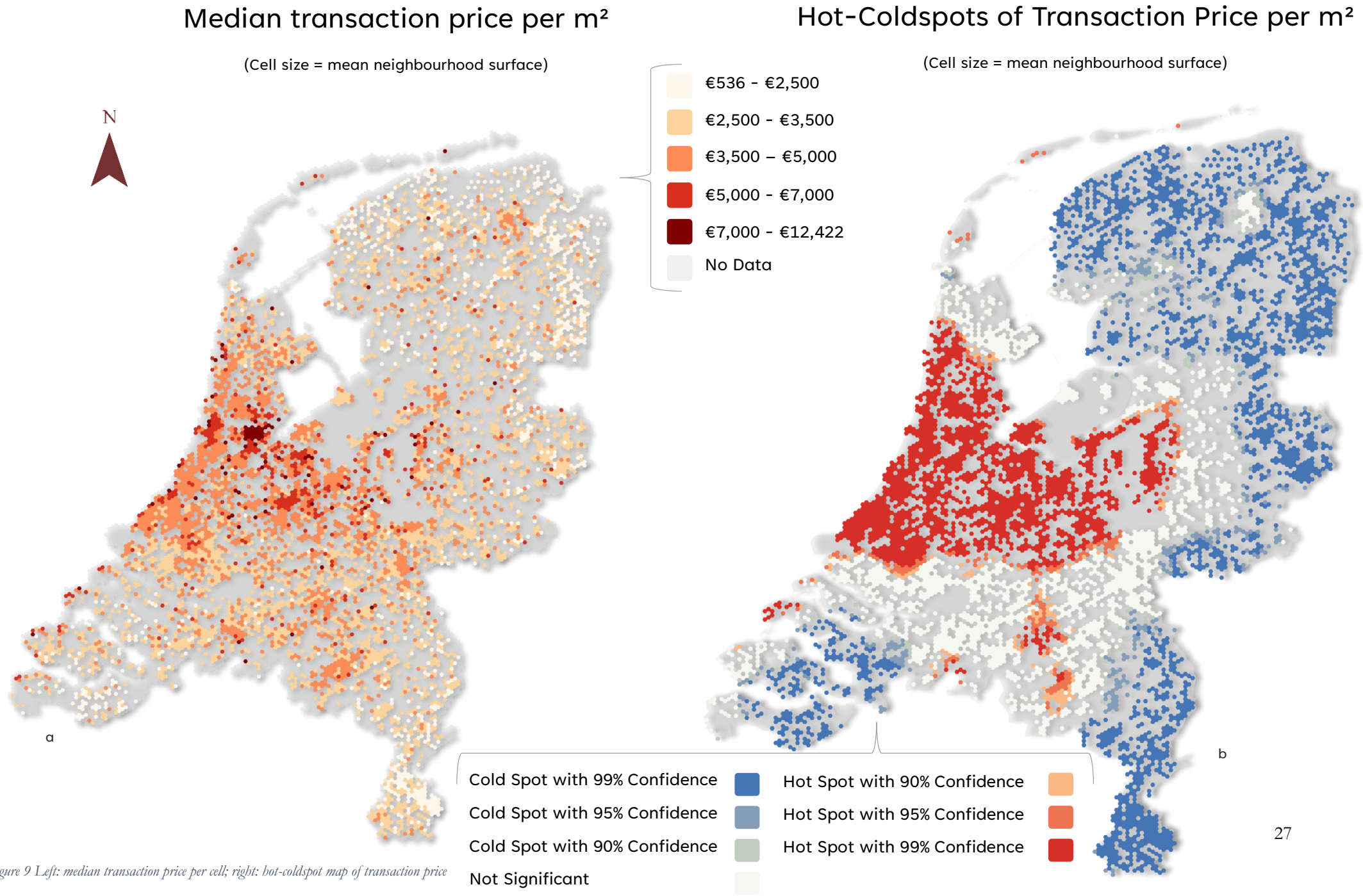



*Figure 9 Left: median transaction price per cell; right: hot-coldspot map of transaction price*

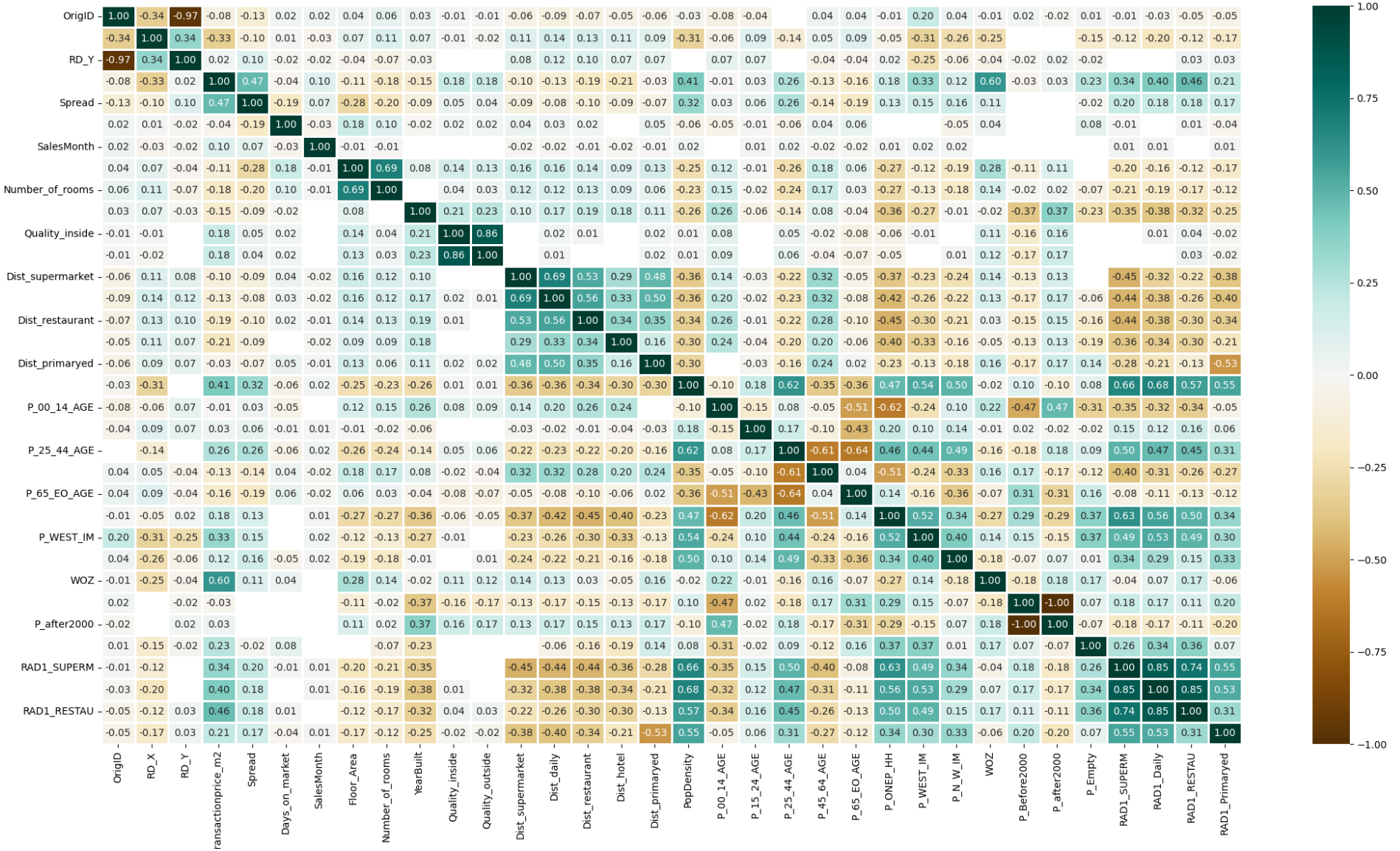



*Figure 10 Correlation matrix of the correlations between the independent and dependent variables*

## 3.7 Analysis

The analysis for this study will be done in four steps. Section 4.7.1 discusses the metrics used to measure the performance of the model and of the individual transactions. Section 4.7.2 discusses the methods used to find out how and where spatial autocorrelation occurs in the prediction accuracy. Following this section, subsection 4.7.3 shows the methodology used to assess the spatial distribution of the feature importance. Finally, section 4.7.4 discusses the steps taken for the spatial regression to find out the factors that influence the prediction accuracy.

### 3.7.1 Prediction Error

The first section of the methodology starts with finding out how we can measure the accuracy of individual transactions. This metric is essential to the study as it will determine the degree of misprediction in the model. It is chosen to use the percentage difference between the transaction price and the predicted price as a measure of accuracy. The benefit of this method is that high and low valued transactions are more easily comparable to each other as opposed to using the absolute difference. A downside of this metric is that it does not measure the deviation from the total average of transactions.

To counter this, the symbology in the maps generated for this section will include a cut-off point for the average prediction error. This allows the reader to see where transactions are inaccurately predicted compared to the average error. However, this requires that the prediction error is absolute.

On the other hand, the prediction accuracy can be either positive or negative. This implies that the map will show both underpredictions and overpredictions. Due to these two reasons, two separate maps are generated to on the one hand, show under and over predicted values and on the other hand, show the prediction error compared to the average prediction error.

To test the different ML algorithms, there are six distinct metrics to assess the accuracy. The first of which is the Mean Squared Error (MSE)(Hodson et al., 2021). This metric measures how much the datapoints are dispersed around the central mean, or in other words, is more consistent. A lower value therefore means that the estimator, or model performs more accurately over all predictions. The Root Mean Squared Error (RMSE) is a general purpose error metric for numerical predictions. According to (Christie & Neill, 2022) the metric is mostly used to compare forecasting errors of different models or model configurations. RMSE is measured to indicate the average deviation of the estimates from the predicted values (Yilmazer & Kocaman, 2020). The Mean Absolute Error (MAE) shows the absolute error value and is easily interpretable (Schneider & Xhafa, 2022). The benefit of using MAE is that the score increases linearly with the increase in errors, meaning that larger errors will show up as an increasingly larger MAE score. The Mean Absolute Percentage Error is another easily interpretable metric, it differs from the other metrics in the fact that it displays the score as a percentage (Kim & Kim, 2016). This percentage shows the magnitude of the errors percentage wise. Furthermore, there is the Median Absolute Percentage Error. This metric works in the same way as the MAPE, however as this metric operates using the median, is less sensitive to outliers

(Hyndman & Koehler, 2006). The final metric $R^2$ measures a goodness of fit and is widely used as a method to determine the variance in de dependent variable that can be explained by the predictor variables. The calculation of these metrics will be done through a python script. The inputs for these metrics are the dependent variables from the test dataset and the predicted values.

### 3.7.2 Emerging spatial patterns

Sub question 2 is the first step into understanding the spatial patterns of the prediction accuracy. This section deals with computing the degree of spatial autocorrelation for the predicted error. As stated spatial autocorrelation can show the presence of spatial variation in a variable. For this section of the research two distinct methods will be used to show the presence of spatial autocorrelation. Firstly, the Moran's I will be generated using the spatial statistics tool in ArcGIS. This value will be an addition to the current metrics that measure the performance of the model. Using this information will give us insight into whether clustering occurs in the dataset. Using this metric will help later on when running a spatial regression. Secondly, hot-cold spot maps will create a better understanding of the clustering patterns. All features are gathered into a hexagonal map using their median value. The median is utilized in this process to eliminate the effect of any outliers. Subsequently, the feature layer is cleared of hexagons with less than three instances. The outcome of this procedure is displayed in the following map represented in Figure 11.

Aggregating the datapoints allow us to make a map of the distribution of the prediction error for each model. While showing the aggregated feature importance gives us an insight in the strongness of the values, it is not ideal for analysing the patterns that occur on a national scale.

**Hot spots & Morans' I**

The Getis-Ord Gi* statistic is a spatial analysis technique employed in GIS to locate areas of spatial clustering. It calculates the spatial autocorrelation of a set of features and identifies regions where values are either similar or dissimilar to their neighbouring areas. If the Getis-Ord Gi* statistic has a positive value, it indicates spatial clustering or hot spots, while a negative value implies spatial outliers or cold spots. In addition to running the Getis-Ord Gi* statistic on the data, the Morans I statistic is used as an additional measurement of performance. The Morans' I value is a value given on the whole of a dataset. The value falls between -1 and 1, where -1 means that the values are perfectly dispersed and 1 meaning that values are perfectly clustered. A value of 0 means that the data is perfectly random, meaning that the values don't have a spatial relation.

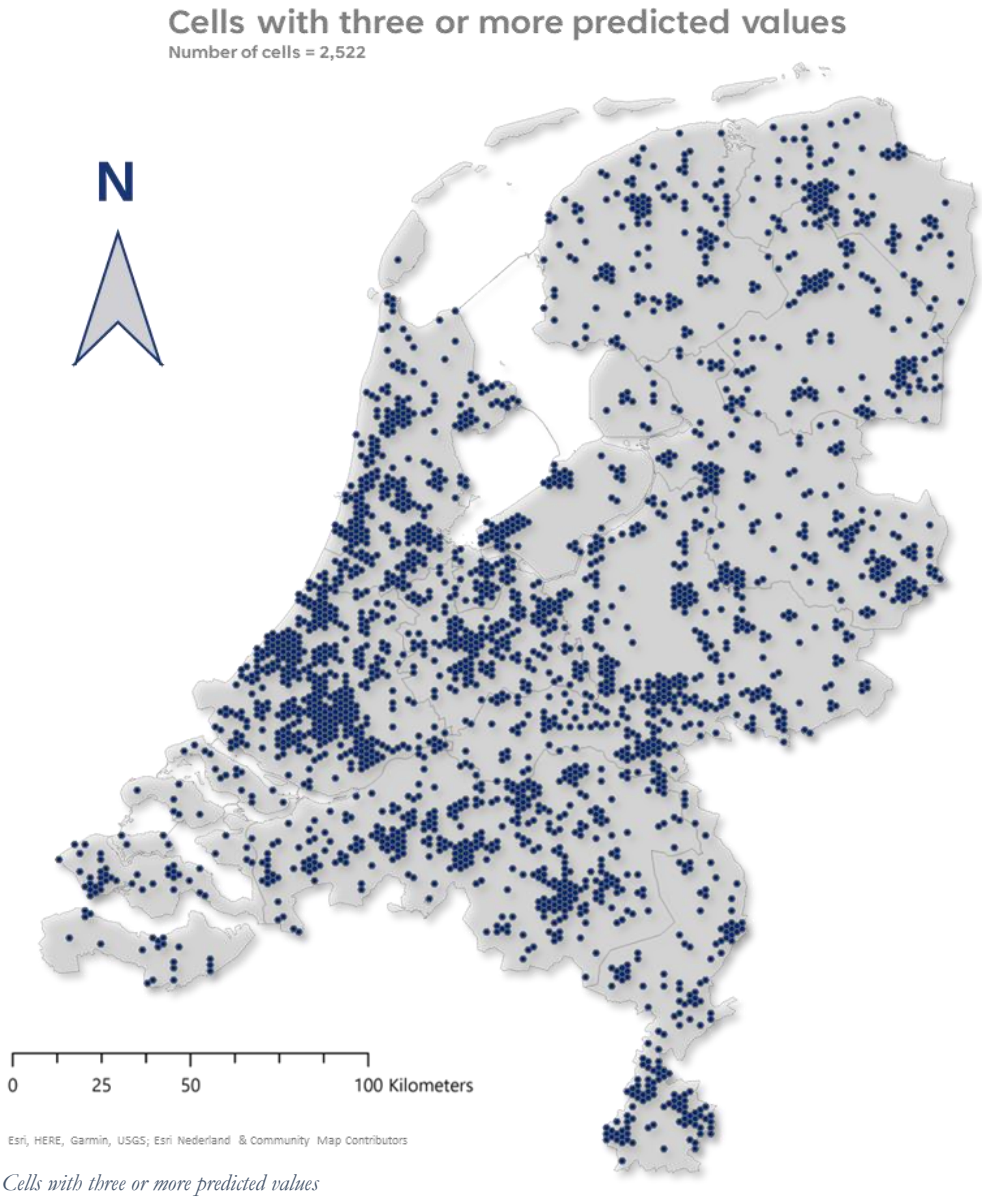



*Figure 11 Cells with three or more predicted values*

### 3.7.3 Feature importance, mapped

Section 3 is the first step in understanding how the different machine learning algorithms take various features into account when generating a prediction. To answer the third question we will once again make use of the hot and cold spot maps to enhance spatial clusters and patterns. All features with a feature importance value higher than one percent are selected for this part of the research. It was deemed too insignificant to assess the features that fall below this threshold. To answer the question of how different machine learning outputs compare in terms of spatial patterns the research presents a set of hot-cold spot maps to assess the importance of values in different places throughout the country. A feature layer with hot-cold spot maps was generated using the ArcGIS Model Builder and the Getis-Ord Gi* statistic. The model for this part is found in the appendix.

The model performs a Hot Spot Analysis on the aggregated dataset for the fields that contribute to more than 1% of total average feature importance.

First The model runs multiple iterations of the Hot Spot Analysis, with different input feature classes (*BuurtHexMean_DTR_JCmin3*, *BuurtHexMean_RF_JCmin3*, and *BuurtHexMean_GB_JCmin3*) and different values of the input field (Input_Field). The input field is dependent on the list of features from the *BuurtHexMean_RF_JCmin3*. The fields from this shapefile is similar to the other two, this means that calling just these fields will suffice for all three algorithms. The input field takes on the value from the iterate fields tool, which iterates over the list of meaningful features. The inputs for these maps were determined using a fixed distance method using a distance band of 15,000 meters. The results of each iteration are stored in a different output feature class, with a name constructed using the input field value (*Hotspot_DTR_%Input_Field%.shp*, *Hotspot_RF_%Input_Field%.shp*, *Hotspot_GB_%Input_Field%.shp*). The string between the percentage signs changes with every iteration. The model for this procedure is documented in the annex (annex D).

**Feature importance & Shapley values**

As stated, the tree models themselves are relatively complicated to understand once they are created. Shapley values increase the level of understanding that can be gained from a model. The values are a measure of assigning significance to individual features in a machine learning algorithm. They measure the contribution of each feature compared to the average predicted transaction price to the overall prediction accuracy of the model. Shapley values are calculated by taking into account all possible combinations of features and their interactions, and assigning a score to each feature based on its contribution to the overall accuracy (Figure 12).

**Shapley values example**

Assume an AVM predicted value for a detached house of €650,000. The house is constructed in 1951 and has a usable floor area of 105m². This house is located in the city of Utrecht in a neighbourhood rich with restaurants. The average price prediction for a house in the whole dataset is about €440,000. This means that the difference in price prediction for the house in Utrecht compared to the average price is €210,000. The Shapley values can explain this difference by looking at the different features. In this case, the fact that the house is in Utrecht adds about €110,000 to the average price. Given that the house is detached from its neighbours adds another €90,000. Additionally, the fact that the house is located in an area with restaurants add €40,000. However, the bad state of the house decreases the price with €30,000. When you add all the contributions to the average, you will end up with the predicted price for this house.

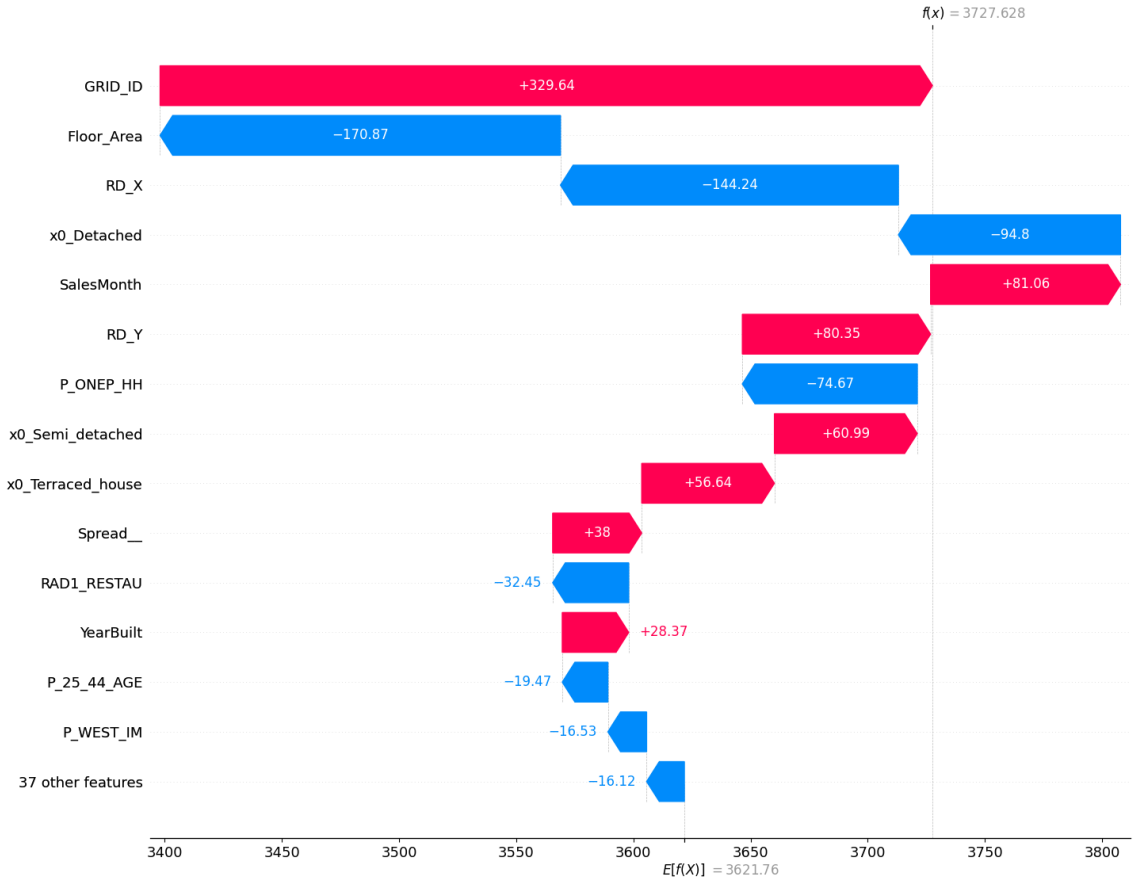



*Figure 12 waterfall plot of example shapley values*

They are calculated by taking the average marginal contribution of each individual or group to the overall outcome. This is done by considering all possible combinations of individuals or groups and their contributions, and then averaging the marginal contribution of each individual or group across all possible combinations.

Feature importance is a measure of how important a feature is in predicting the outcome of a machine learning algorithm. It is calculated by looking at the relative contribution of each feature to the overall accuracy of the model. Features with higher importance are more likely to be used in the model and have a greater impact on the accuracy of the model. Shapley values and feature importance are both important measures for understanding the performance of a machine learning algorithm. Shapley values provide an insight into how much each feature contributes to the overall accuracy, whereas feature importance provides an indication of which features are most important for the model. By understanding both of these measures, data scientists can better understand the performance of their models and make informed decisions about which features to include or exclude from their models.

For this research the relative contribution of the Shapley values are calculated to assess the importance of the features for one prediction. The feature importance for every instance is then used to create aggregated maps to show the spatial patterns in the importance of these values.

### 3.7.4 Spatial regression model

Spatial regression will be used as a method to find out which features have the most impact on the prediction error in this study. The guide on spatial regression in GeoDa written by Anselin (2005) provides a good framework for carrying out spatial regression. The schema for determining the best spatial regression method is added in the annex (annex E). The model dictates that a regular OLS regression needs to be performed at first. In order to determine whether a spatial model would fit the data better, we need to look at the diagnostics. A few factors are of major importance. These are the multicollinearity condition number, diagnostics for heteroskedasticity and diagnostics for spatial dependence (Bell & Owusu, 2021). After considering these factors, the spatial lag and the spatial error models were run, to see if they fit the data better. For this the $R^2$ and the adjusted $R^2$ are good metrics to consider. During the initial phase of the regression, it was found out that the independent variables correlate with each other, so much so that the multicollinearity condition number exceeded 30. It is therefore considered best practice to revaluate the model in order to lower this number. This was deemed impossible as there would be only two variables left in the model. A solution to this problem was to use a regression technique that doesn't take multicollinearity in consideration. Random Forest regression, was deemed a reasonable and quick alternative to rank the independent variables in terms of their importance to the prediction error. The same method of feature importance was used to assess the importance of each independent variable.

# 4

# Analysis and Results

The analysis and results chapter focusses on presenting the outcomes of the assessments of the individual transactions as well as the algorithms in general. Section 5.1 starts by showing the metrics for the three algorithms in terms of performance. In addition to this, the most important features are selected for use in the next sections of the research. Section 5.2 continues with presenting the spatial patterns in prediction accuracy. The maps shown in this section will provide the reader with a clear understanding of where the prediction accuracy in the highest. Section 5.3 presents multiple hot-cold spot maps of the most important variables in all algorithms. These maps are extended with a brief interpretation. Not only, were there similar patterns visible among the algorithms, but also among different variables. The final section (section 5.4) provides the outputs of the regression analysis. In addition to this, a list of most important features that might impact the prediction accuracy are presented according to a Random Forest regression.

## 4.1 Performance metrics

**Model performance**

When assessing the performance metrics in Figure … we can observe that the Random Forest algorithm performs best compared to the Decision Tree and Gradient Boosting methods. This is most obvious in Figure 13a. The mean absolute error metric shows that the RF method has a mean absolute error of €387.77, this translates to a mean deviation of 11.2% (Figure 13d). When assessing the model fit using the $R^2$ metric we can see that the RF algorithm performs best again. A high $R^2$ means that the model is able to explain most of the variance (Figure 13f).

**Individual transaction performance**

Using the methods of absolute percentage error and percentage error it was possible to map the prediction accuracy of individual transactions. Spatially joining these instances to the hexagon grid shows locations with underpredicted and over predicted transactions (Figure 14). The maps themselves don't show any visible patterns. However, when simply counting the number of over- and underpredictions as well as the number of accurate predictions we can see that the Random Forest algorithm shows the most promising

results. The Gradient Boosting algorithms shows the highest number of underpredicted values whereas the decision tree method shows the highest number of overpredictions.

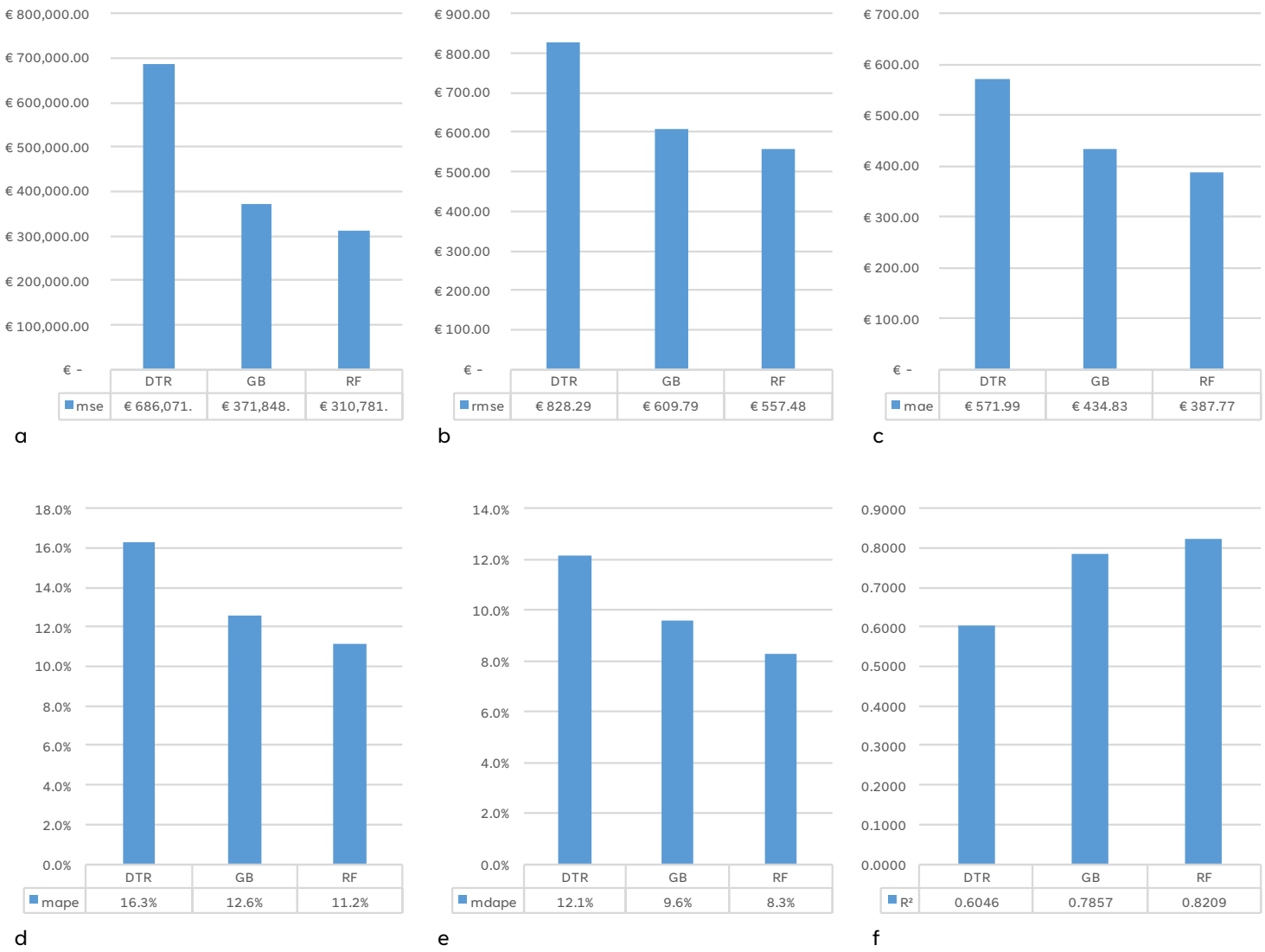



*Figure 13 performance metrics (a: MSE; b: RMSE; c: MAE; d: MAPE; e: MdAPE; f: $R^2$)*

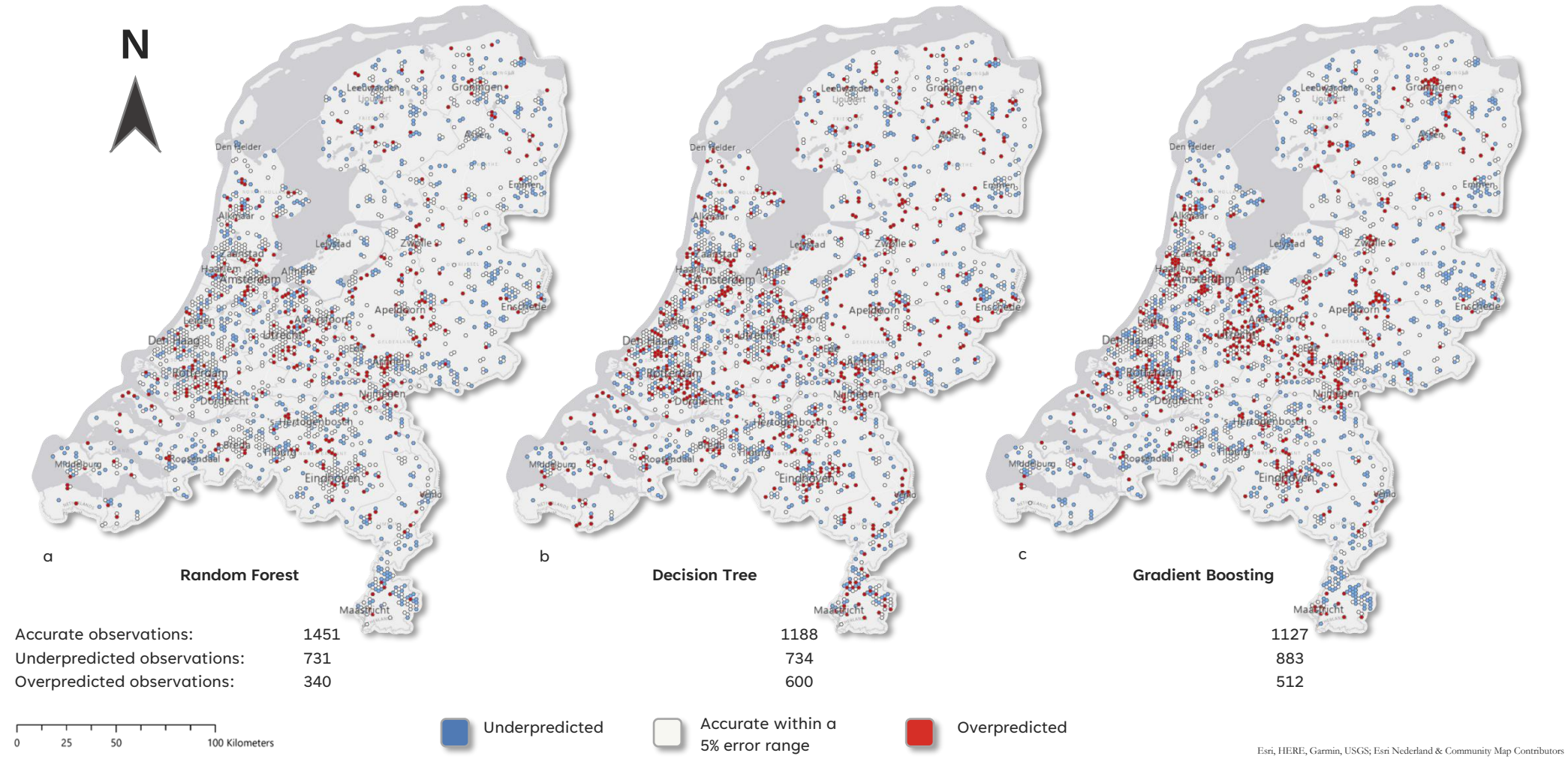



*Figure 14 Over and underpredicted transactions*

These maps portray the cells where predictions were over or underestimated. This map was cleared of grid cells that contained under three predictions. Using these maps, it's impossible yet to recognize any particular patterns. Nonetheless, with the information behind this map we do see a significant difference between the three algorithms in the number of aggregated observations that fall within 5% accuracy range. We can also observe that all algorithms tend to underpredict rather than overpredict.

## 4.2 Patterns of Prediction Accuracy

Using the Getis Ord hot spot analysis, we can see emerging patterns in misprediction in the different algorithms on the maps. The following maps in this section show the outcomes of the hotspot analysis including an interpretation of the maps. In addition to these maps, a Moran's I statistic was calculated to show the magnitude of spatial autocorrelation existent in the prediction accuracy. The chart in Figure 15 shows that the Random Forest algorithm shows the highest Moran's I value compared to the Gradient Boosting and Decision Tree algorithm. Even though the Morans index is small it was highly significant, showing signs of clustering in all three algorithms with regard to the prediction accuracy (absolute percentage error) (table 2). The generated reports for the Moran's I are added to the annex (annex F-H).

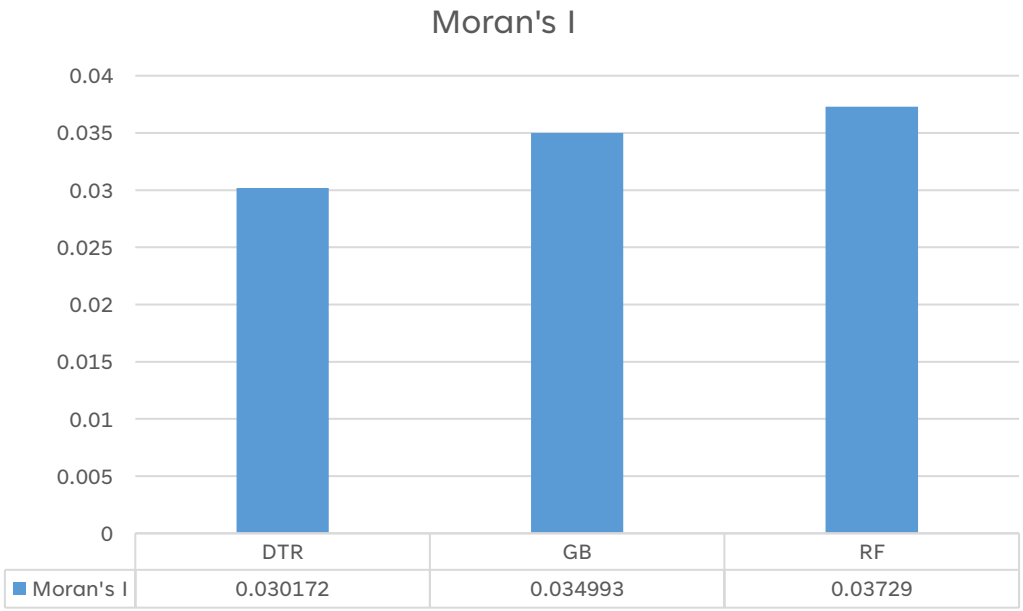



*Figure 15 Moran's I of the absolute percentage error*

*Table 2 Moran's I of the absolute percentage error (including Z-score and significance)*

| Algorithm | Moran's I | z-score | p-value |
|---|---|---|---|
| Random Forest | 0.037290 | 15,841166 | 0.000000 |
| Gradient Boosting | 0.034993 | 14.865374 | 0.000000 |
| Decision tree | 0.030172 | 12.736804 | 0.000000 |

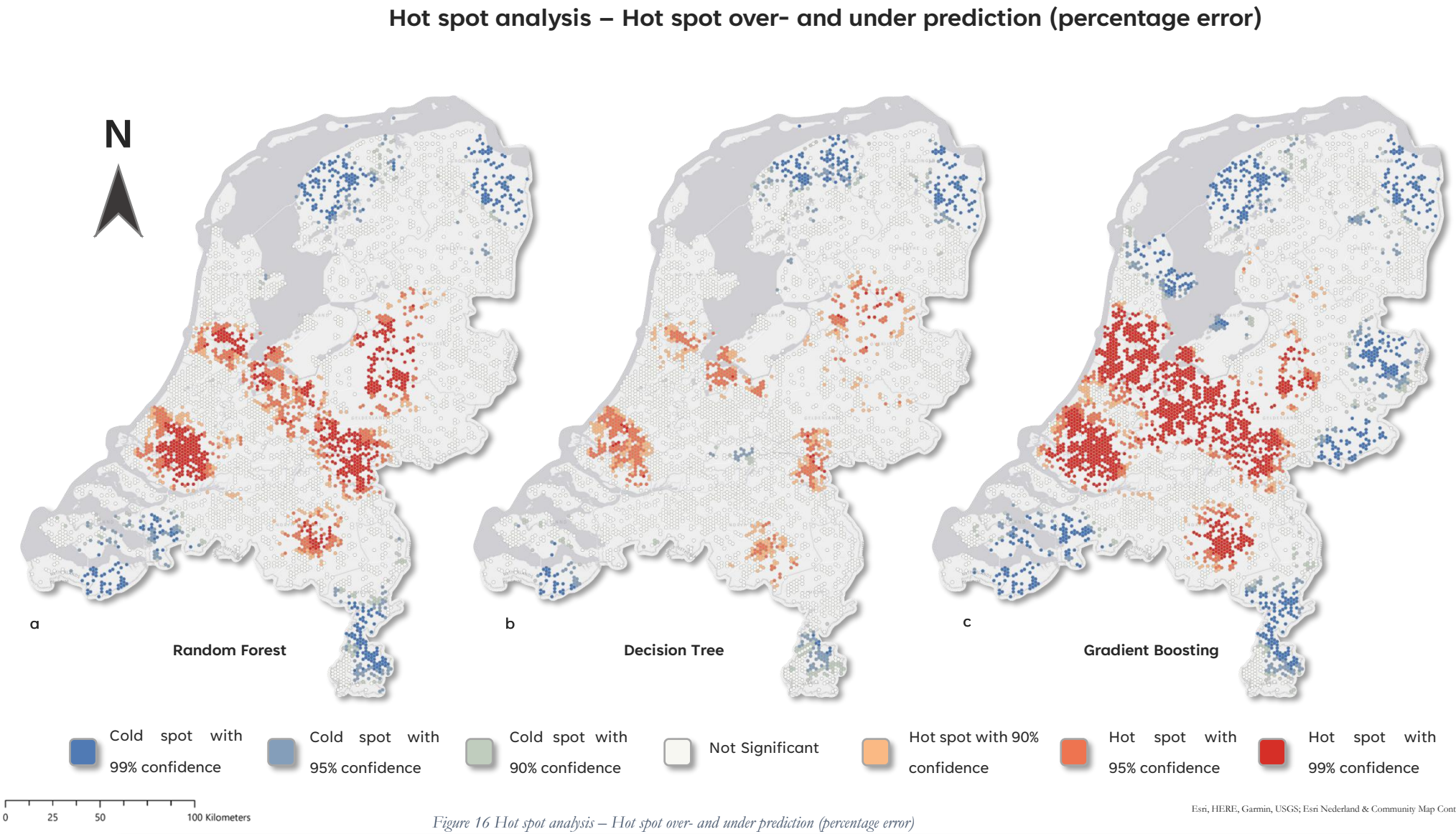



*Figure 16 Hot spot analysis – Hot spot over- and under prediction (percentage error)*

To be able to distinguish any patterns in the maps from Figure 14 the maps in Figure 16 are used. The red zones show a clustering of overpredicted values whereas the blue zones show underpredicted areas. We can observe from these maps, that the overpredicted locations are mostly in the more densely populated areas, in and around the Randstad. While the RF and GB algorithms show more significant hotspots, the DTR method show smaller hotspots with less significance.

## Hot spot analysis – prediction accuracy (absolute percentage error)

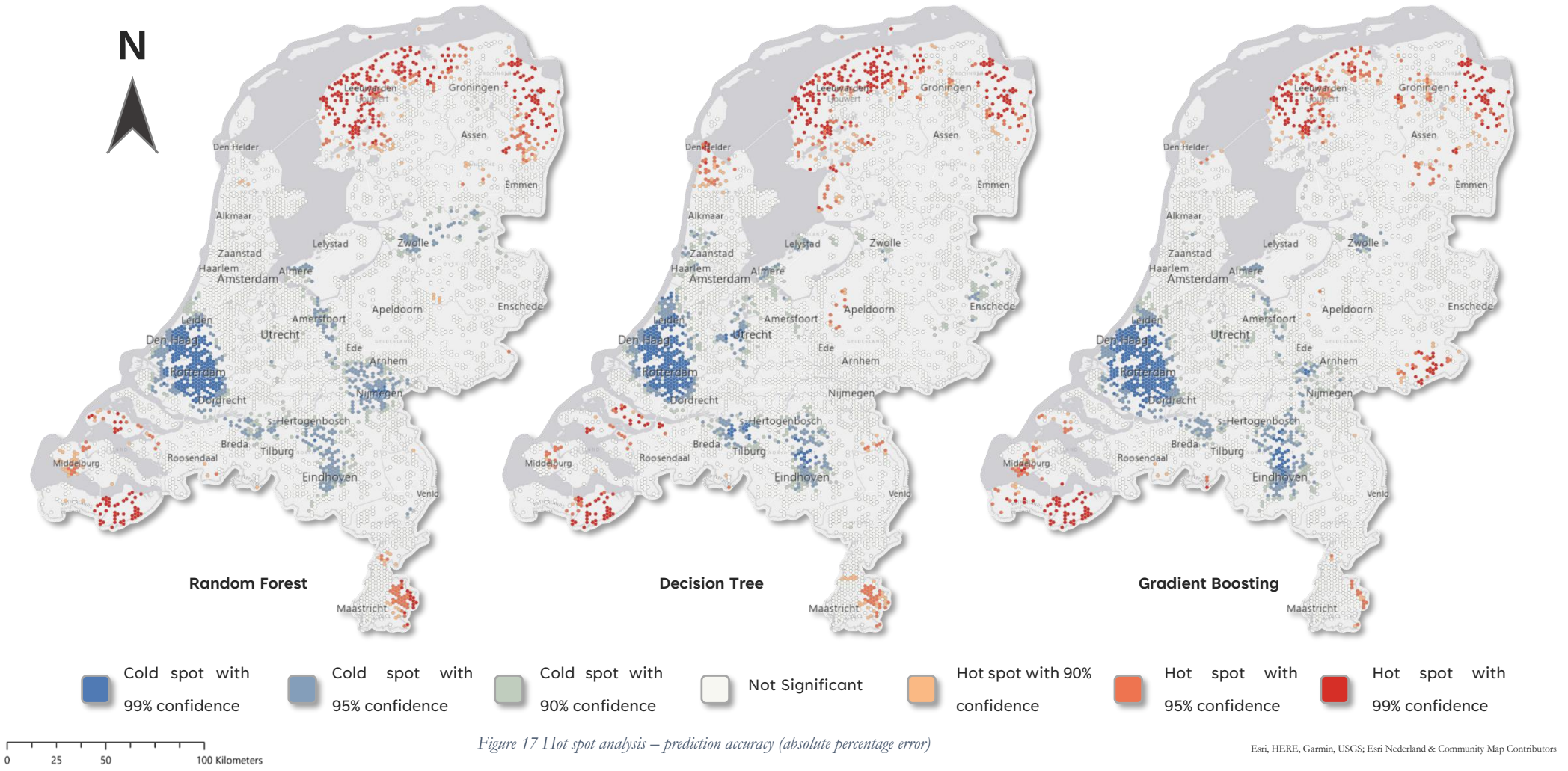



*Figure 17 Hot spot analysis – prediction accuracy (absolute percentage error)*

Where the previous maps tell us something about the direction of the prediction accuracy, these maps show where the instances are more accurately predicted than others. All three algorithms show cold spots in and around Rotterdam, Breda, Eindhoven, Almere and Zwolle, meaning that the accuracy is highest in these areas. All three algorithms show similar patterns when it comes to hotspots. Most of the locations with the higher absolute percentage error are located on the edges of the country. Especially in the province of Friesland there seems to be a high level of misprediction.

## 4.3 Spatial manifestation of feature importance

The feature importance for the overall model was calculated using the impurity based feature importance module. This model is used to interpret the importance of the individual features in the model and is designed for tree based algorithms. Even though this model tends to be biased towards high cardinality variables, the outcomes show the importance of a house being detached (table 3; Figure 18).

*Table 3 most important features in each model*

| Labels | Sum of RF importance | Sum of GB importance | Sum of DTR importance | Sum of AVG percentage |
|---|---|---|---|---|
| WOZ | 32.09% | 36.90% | 31.18% | 33.39% |
| GRID_ID | 11.17% | 14.26% | 11.66% | 12.37% |
| PopDensity | 6.85% | 8.18% | 12.29% | 9.11% |
| Floor_Area | 8.36% | 8.01% | 8.89% | 8.42% |
| RD_X | 5.01% | 5.46% | 5.53% | 5.33% |
| RD_Y | 3.91% | 5.39% | 4.27% | 4.52% |
| YearBuilt | 2.80% | 2.67% | 2.87% | 2.78% |
| P_ONEP_HH | 2.51% | 2.97% | 2.48% | 2.66% |
| Spread__ | 3.00% | 1.67% | 2.87% | 2.52% |
| RAD1_RESTAU | 2.54% | 4.50% | 0.50% | 2.51% |
| RAD1_SUPERM | 5.07% | 0.68% | 0.35% | 2.03% |
| P_25_44_AGE | 1.58% | 1.20% | 1.84% | 1.54% |
| x0_Detached | 1.19% | 1.70% | 1.24% | 1.38% |
| Quality_inside | 1.06% | 1.71% | 1.09% | 1.29% |
| P_WEST_IM | 1.11% | 1.49% | 1.04% | 1.21% |
| **Total** | **88.25%** | **96.81%** | **88.11%** | **91.06%** |

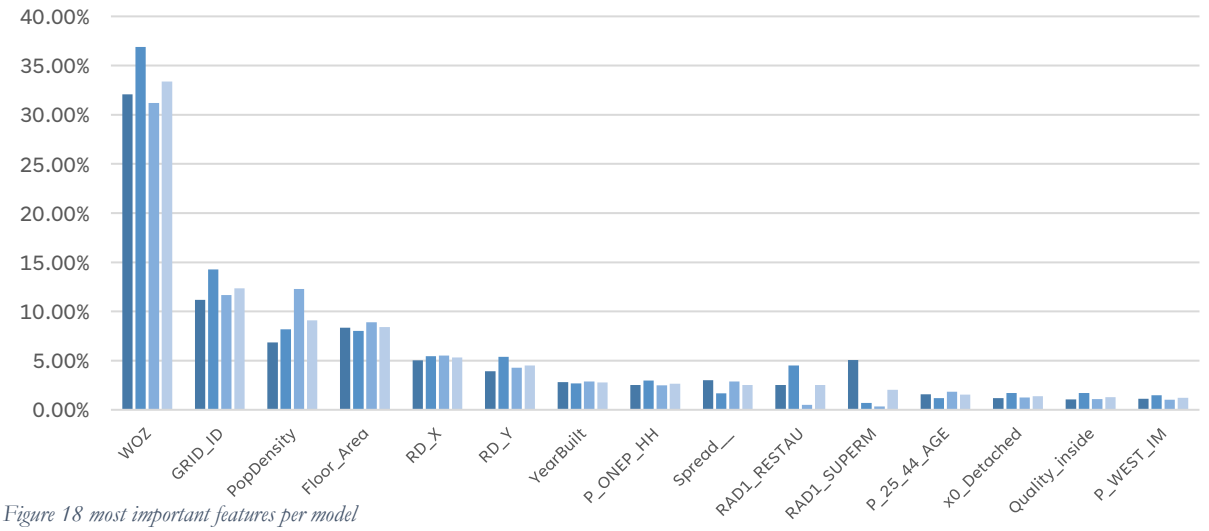



*Figure 18 most important features per model*

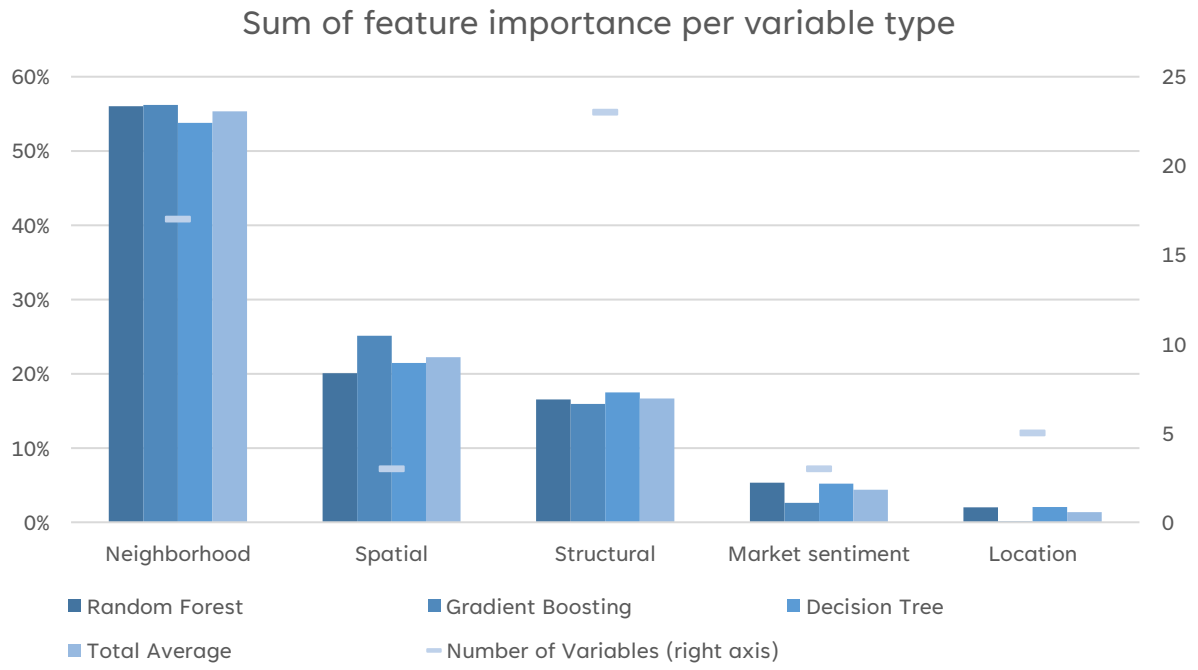



*Figure 19 most important variables per type*

The most important type of variables are the neighbourhood variables (Figure 19). However, we can also observe that the number of variables for this type is relatively high. Considering this, the spatial variables (RD_X, RD_Y and GRID_ID) show high importance. Even though the category of structural variables includes a high number of variables we can see that the importance is lower than the spatial or neighbourhood variables.

This section continues with the presentation of the spatial importance maps of the different variables. Each algorithm shows its own characteristics in the maps. In some maps the feature importance shows similar patterns while in other the hot and cold spots differ.

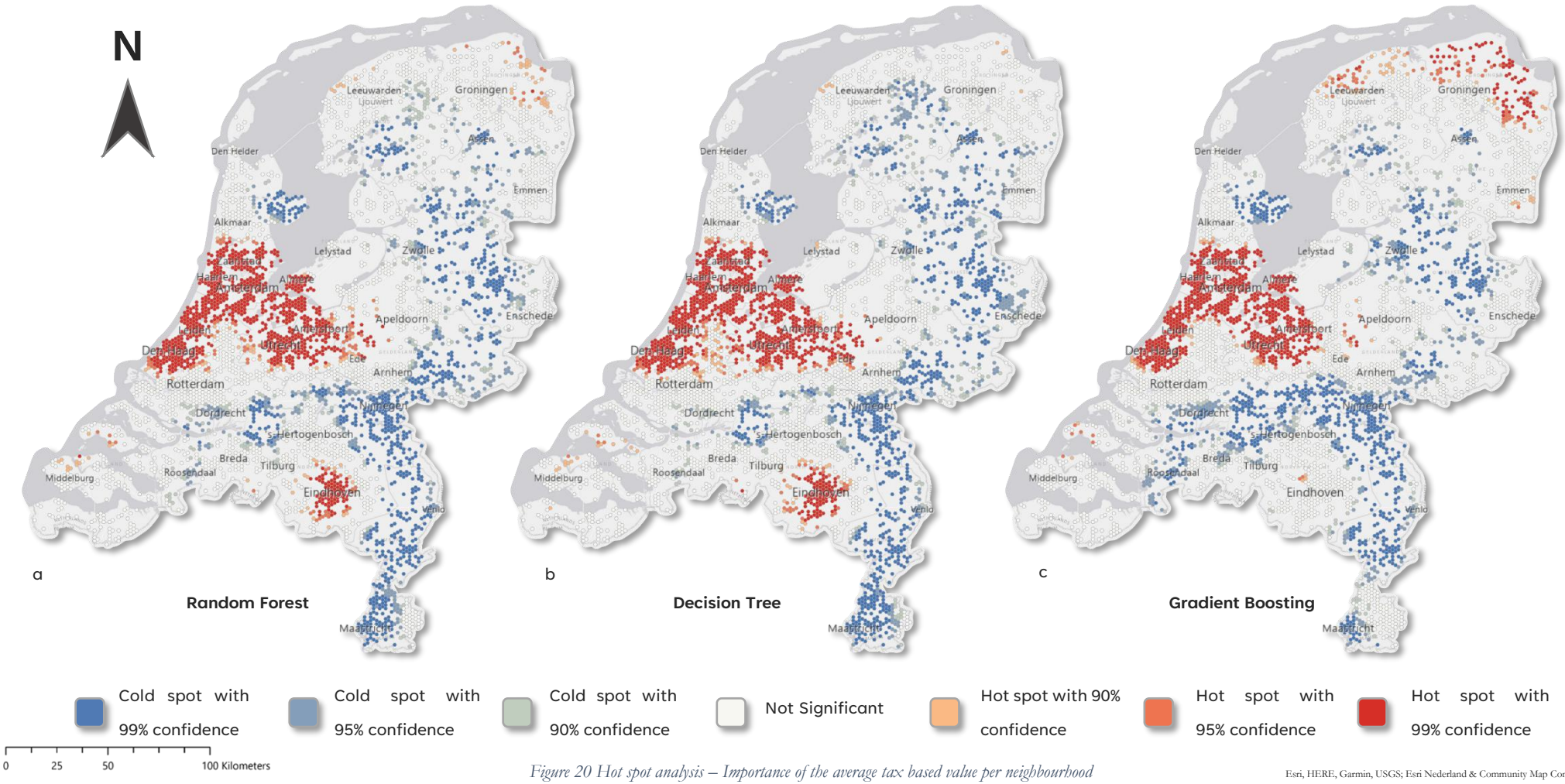



*Figure 20 Hot spot analysis – Importance of the average tax based value per neighbourhood*

Of all features tested, the average tax based value per neighbourhood has shown to be the most important predictor of the housing prices. It plays a more important role in regions with a high transaction price than it does in regions with a lower transaction price. The maps show similar patterns however Eindhoven does not show a particular high importance with regards to this feature.

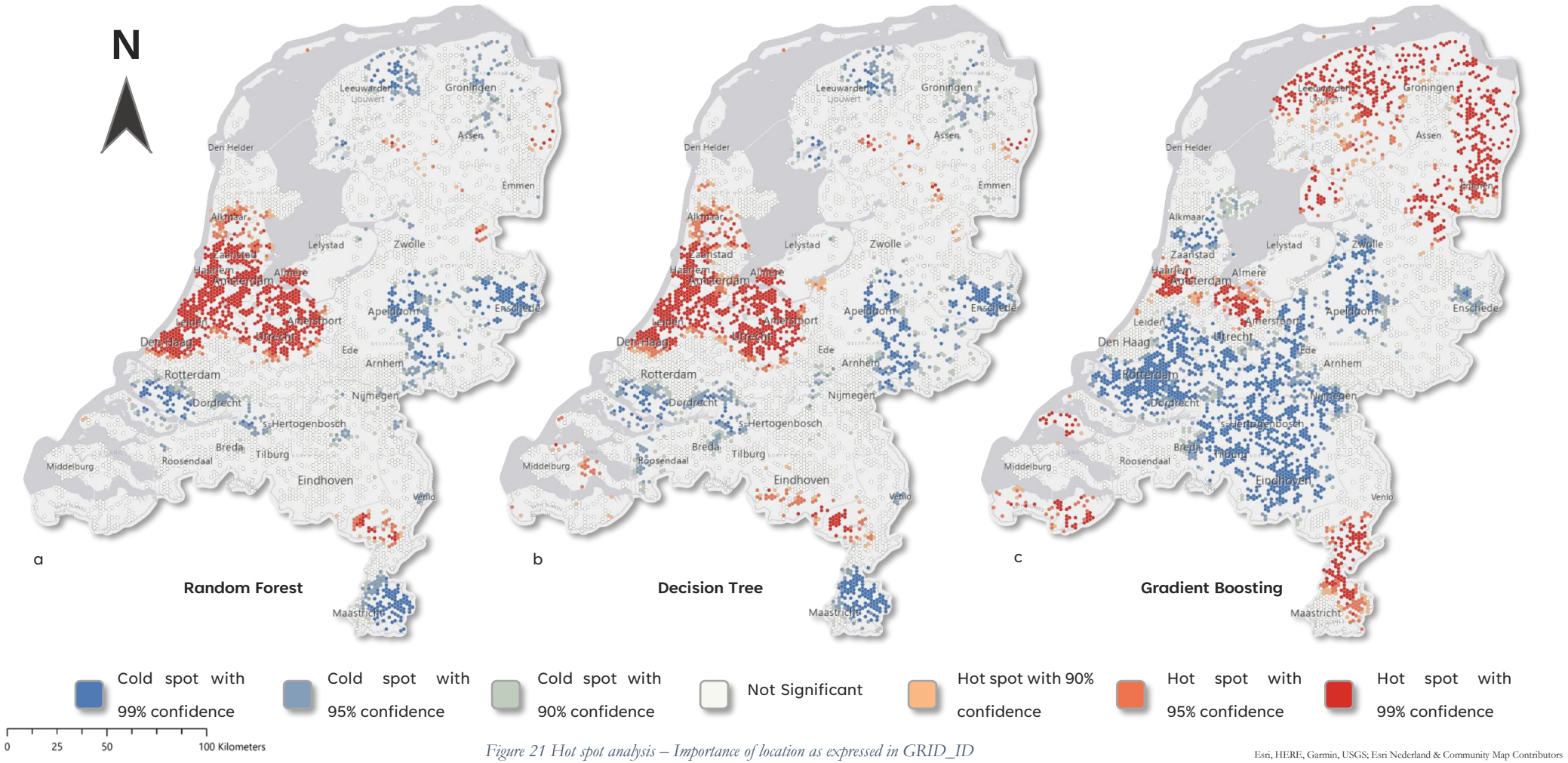



*Figure 21 Hot spot analysis – Importance of location as expressed in GRID_ID*

The maps on this page show the importance of the variable Grid_ID. The fact that a transaction is located in a grid in and around Utrecht and Amsterdam seems to be important to the price prediction according to what these maps visualize. The RF and DTR show similar patterns whereas the GB algorithm shows a pronounced cold spot in the province of North-Brabant, South-Holland and Utrecht. Additionally, this map shows a surprising hotspot in the north of the country. These maps are indicators of where location is a relatively important factor in the price prediction.

## Hot spot analysis – Importance of the Population density

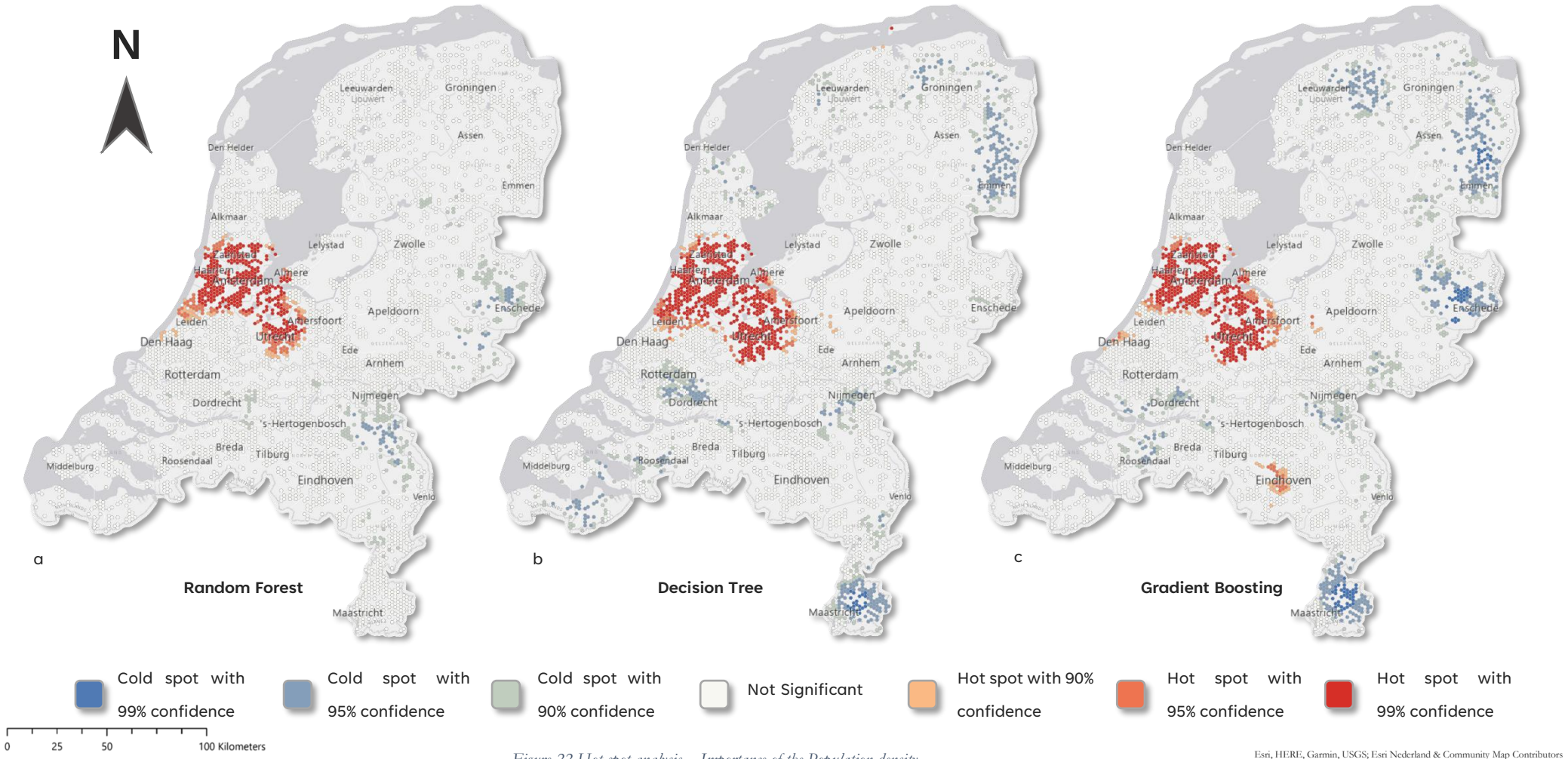



*Figure 22 Hot spot analysis – Importance of the Population density*

The population density plays an important role in these outcomes in the most densely populated areas. In other areas the population density does not seem to play a significant role.

## Hot spot analysis – Importance of the Usable Floor Area

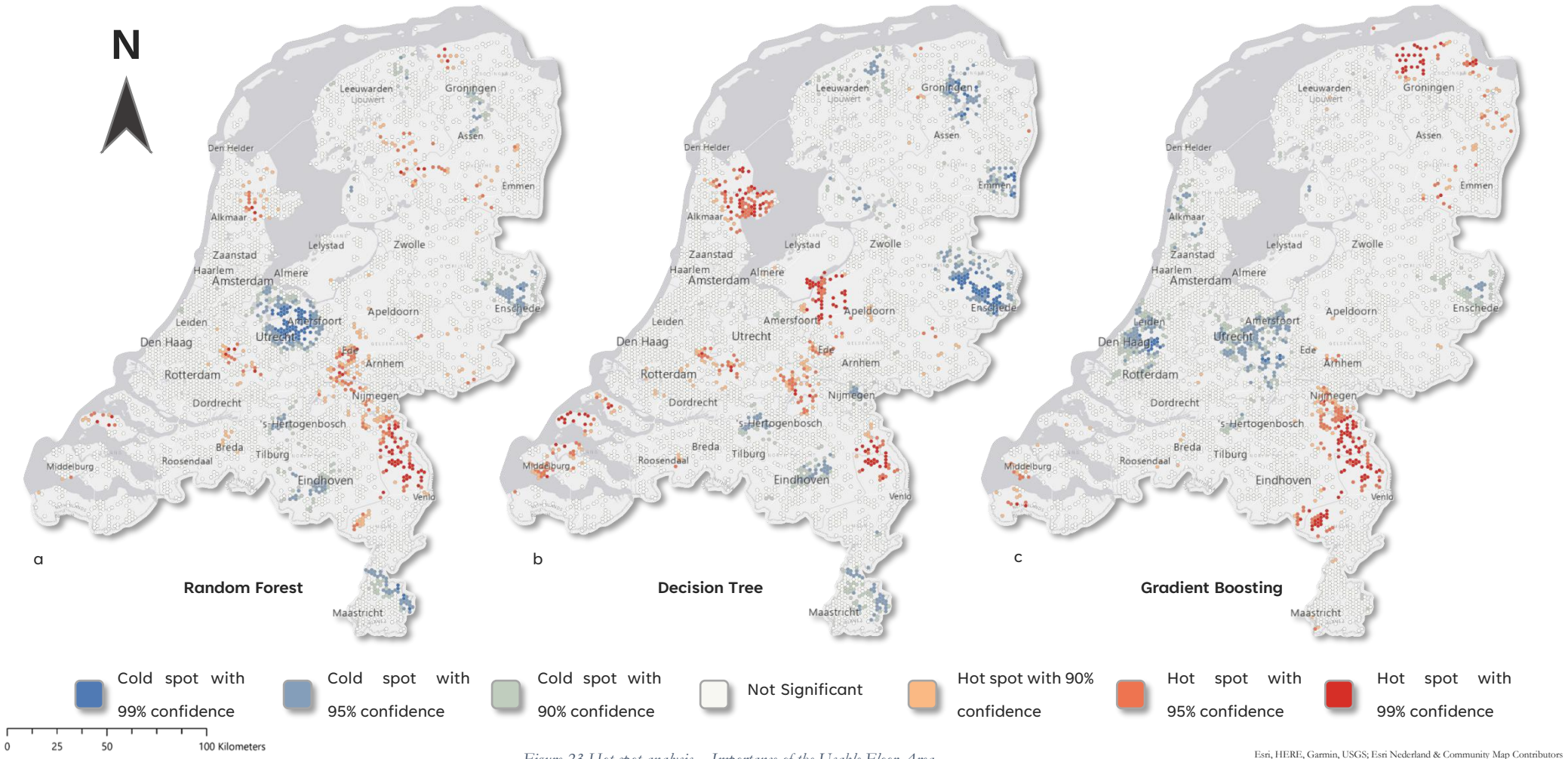



*Figure 23 Hot spot analysis – Importance of the Usable Floor Area*

All three maps show a significant hotspot in the south eastern part of the country, around Nijmegen and Venlo. This shows that the usable floor area was an important factor for the price prediction in 2021 in this location. No other regions show very big hot or cold spots, meaning that the importance is relatively homogenous.

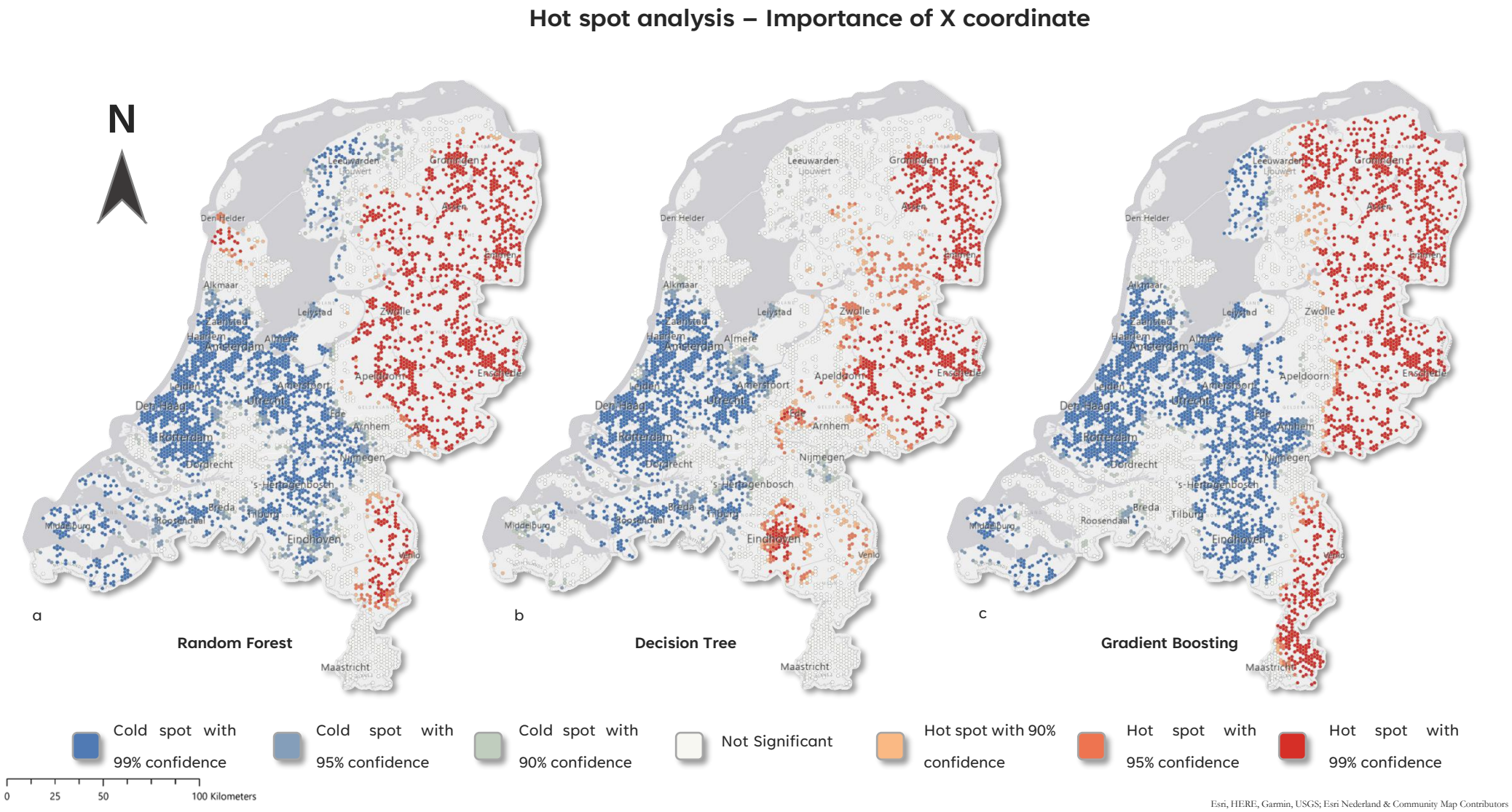



*Figure 24 Hot spot analysis – Importance of X coordinate*

The feature importance of the variable in this map is relatively abstract and hard to interpret. The blue regions in this case mean that the X- coordinate is less of importance than in the red regions. All algorithms show similar patterns on a broad scale, however there is the exception of Eindhoven in the DT algorithm. The results of these maps generally indicate that in 2021, the further you go east, the less important the role of the X-coordinate in the prediction of the price.

## Hot spot analysis – Importance of Y coordinate

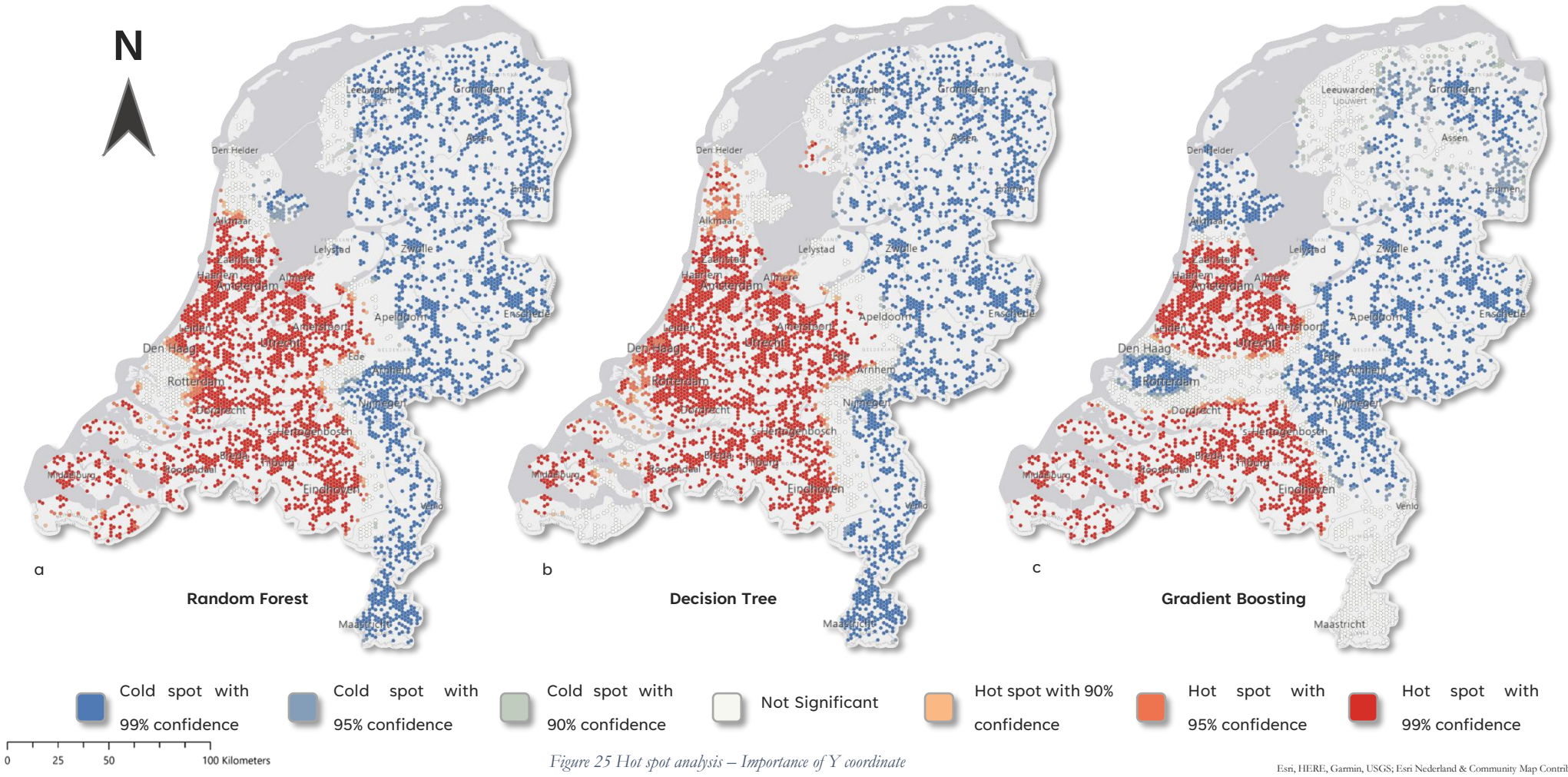



*Figure 25 Hot spot analysis – Importance of Y coordinate*

The feature importance of the variable in these maps are similarly interpreted as the maps displayed in Figure 14. The blue regions in this case mean that the Y-coordinate is less of importance than in the red regions. All algorithms show similar patterns on a broad scale, however there is the exception of Rotterdam in the GB algorithm. The results of these maps generally indicate that in 2021, the further you go east, the less important the role of the Y-coordinate in the prediction of the price.

## Hot spot analysis – Importance of the construction year

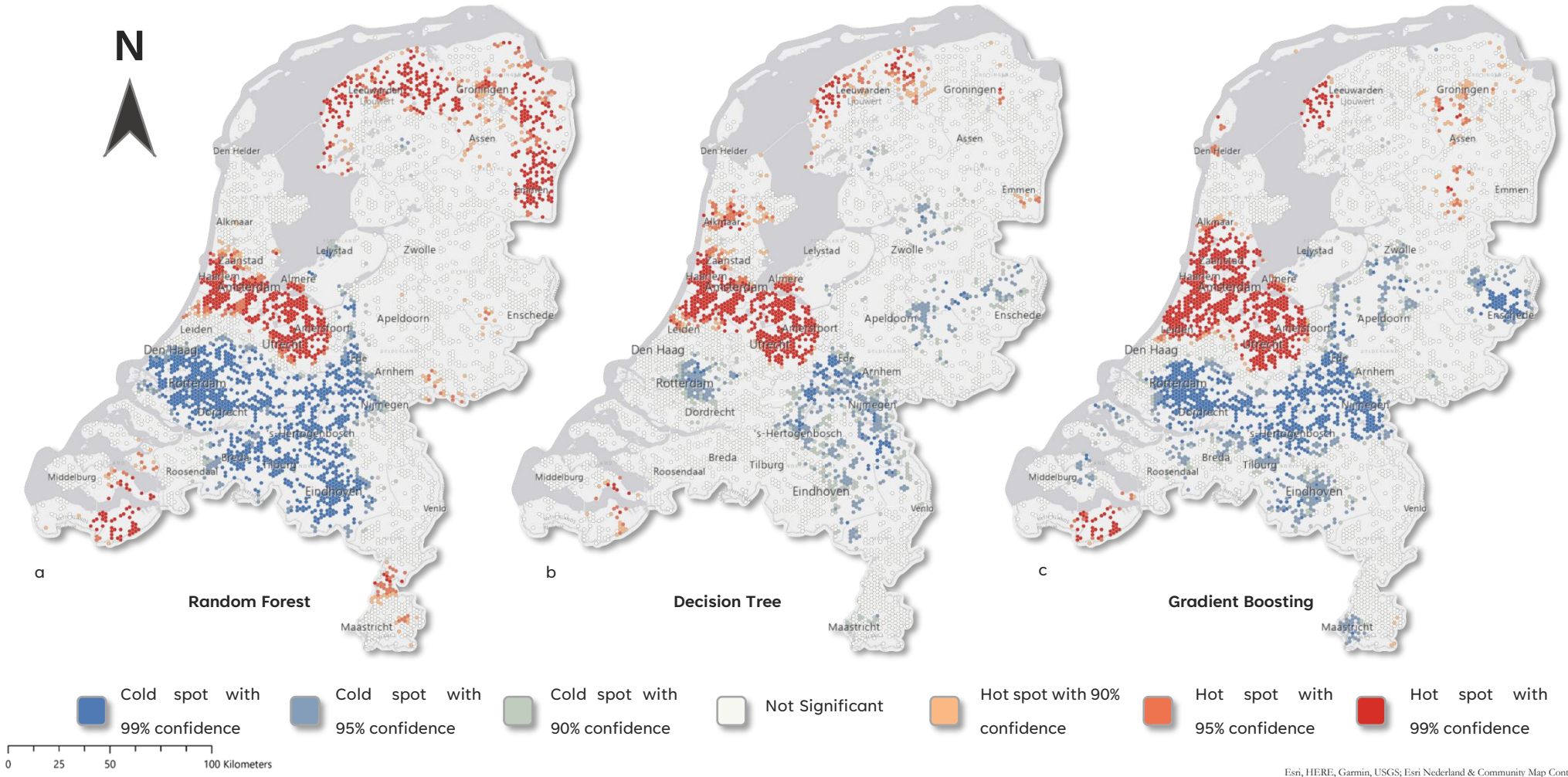



*Figure 26 Hot spot analysis – Importance of the construction year*

The maps on construction year and tax based value show quite similar patterns in terms of hot and cold spots. The majority of the hot spots for the importance of the construction year can be seen in the greater Amsterdam region and around Utrecht. Additionally, Groningen shows up as an important hotspot.

## Hot spot analysis – Importance of the percentage of single person households in a neighbourhood

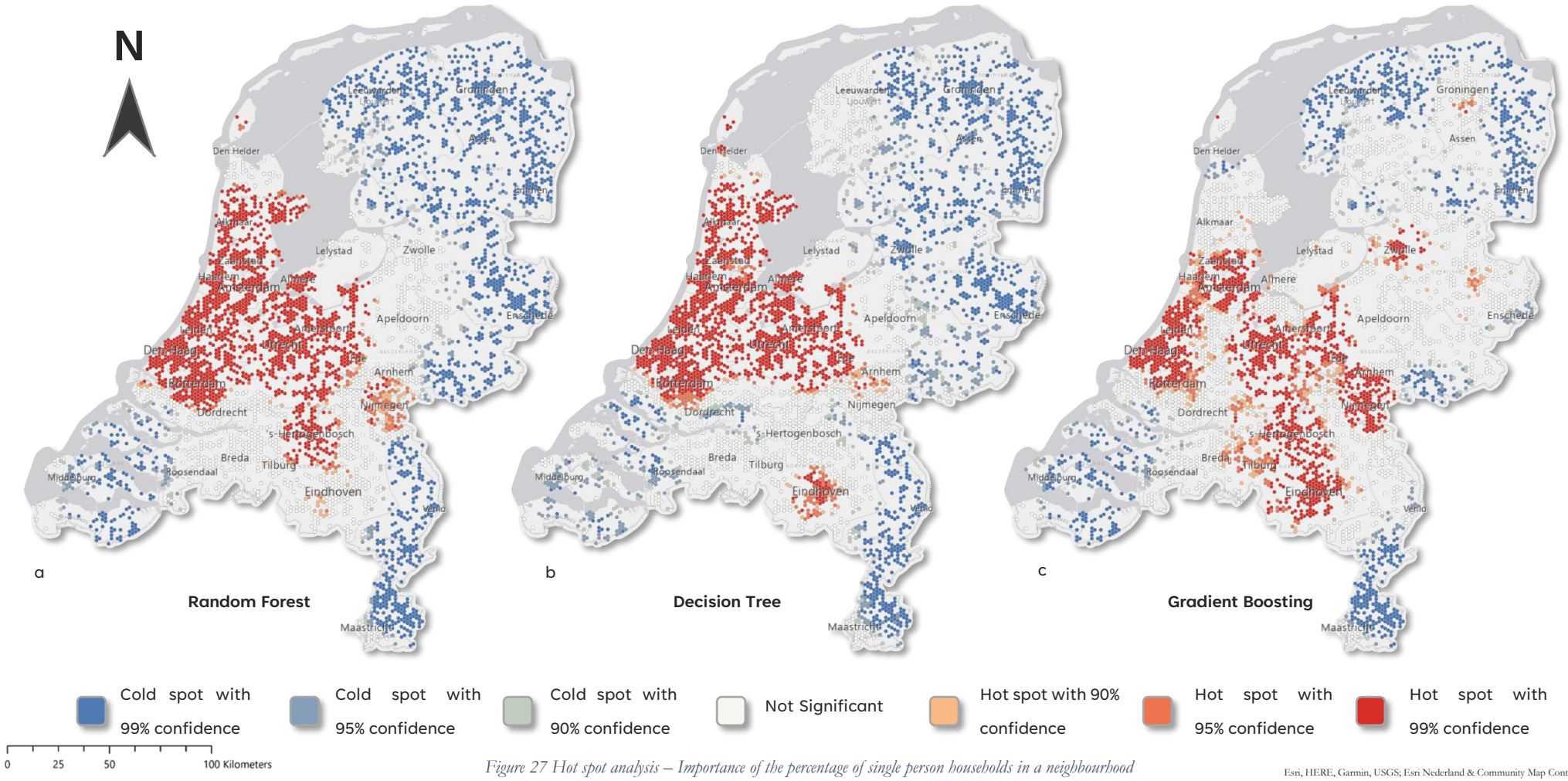



*Figure 27 Hot spot analysis – Importance of the percentage of single person households in a neighbourhood*

In terms of the importance of the percentage of single person households in a neighbourhood we can see that all the algorithms more or less show the same patterns. The hotspots are most significantly present in the Randstad whereas the cold spots mainly show up on the borders. An interesting difference can be seen in the city of Groningen and Zwolle. These cities show up as hotspots in the GB map but don't show up in the other maps. The importance of this variable might be closely related to the scarcity of housing.

## Hot spot analysis – Importance of the difference between the asking price and transaction price

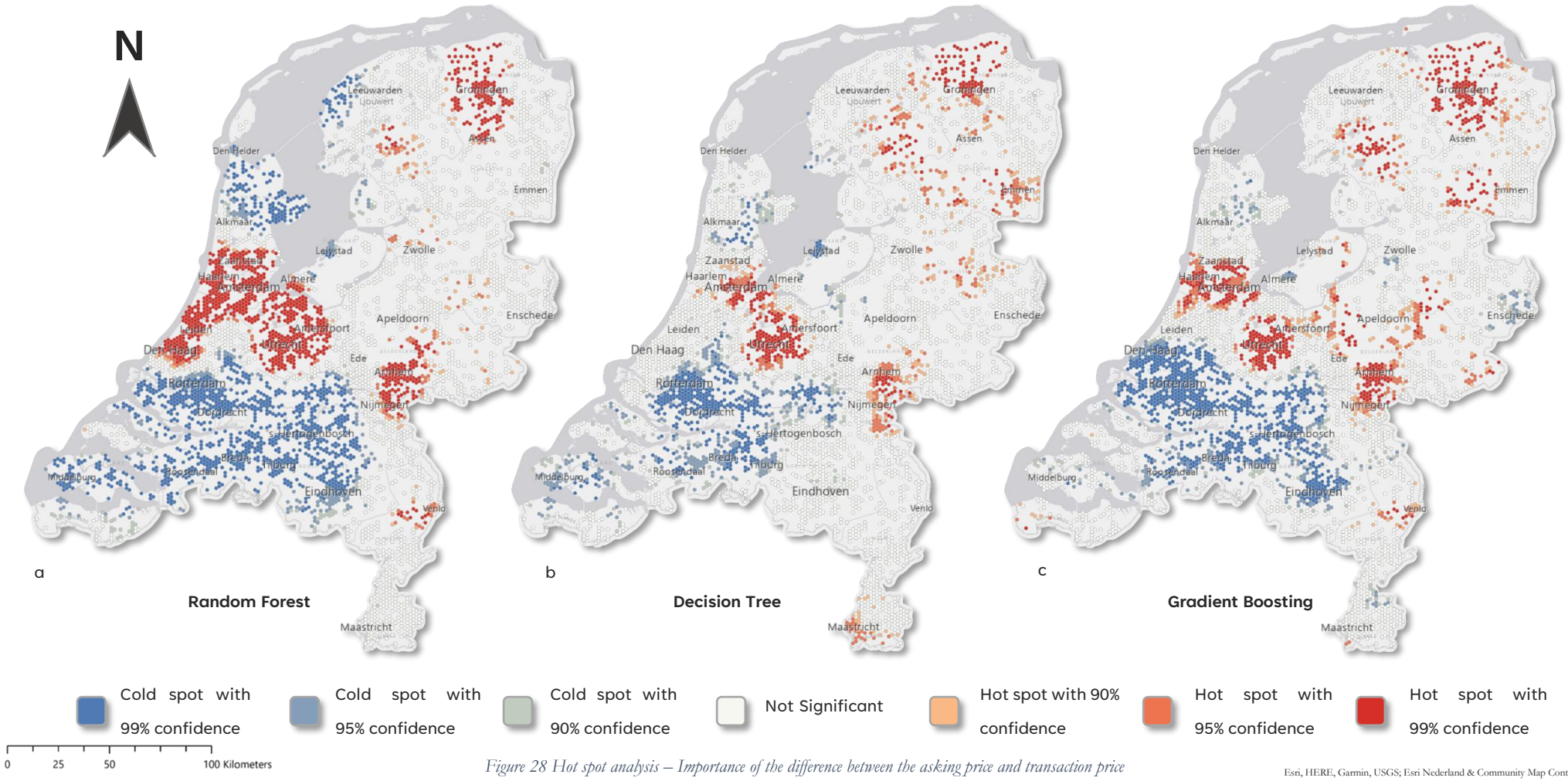



*Figure 28 Hot spot analysis – Importance of the difference between the asking price and transaction price*

These maps show the importance of the factor spread. Spread is defined as the percentage difference between the asking price and the transaction price. We can observe similar patterns in all three maps. Amsterdam, Utrecht, Arnhem and Groningen all show up as hotspots whereas the greater Rotterdam region and most of North-Brabant show signs of lower importance on this variable. One reason for the importance of this factor in these regions might be the difference in available housing.

## Hot spot analysis – Importance of number of restaurants within a one kilometre radius

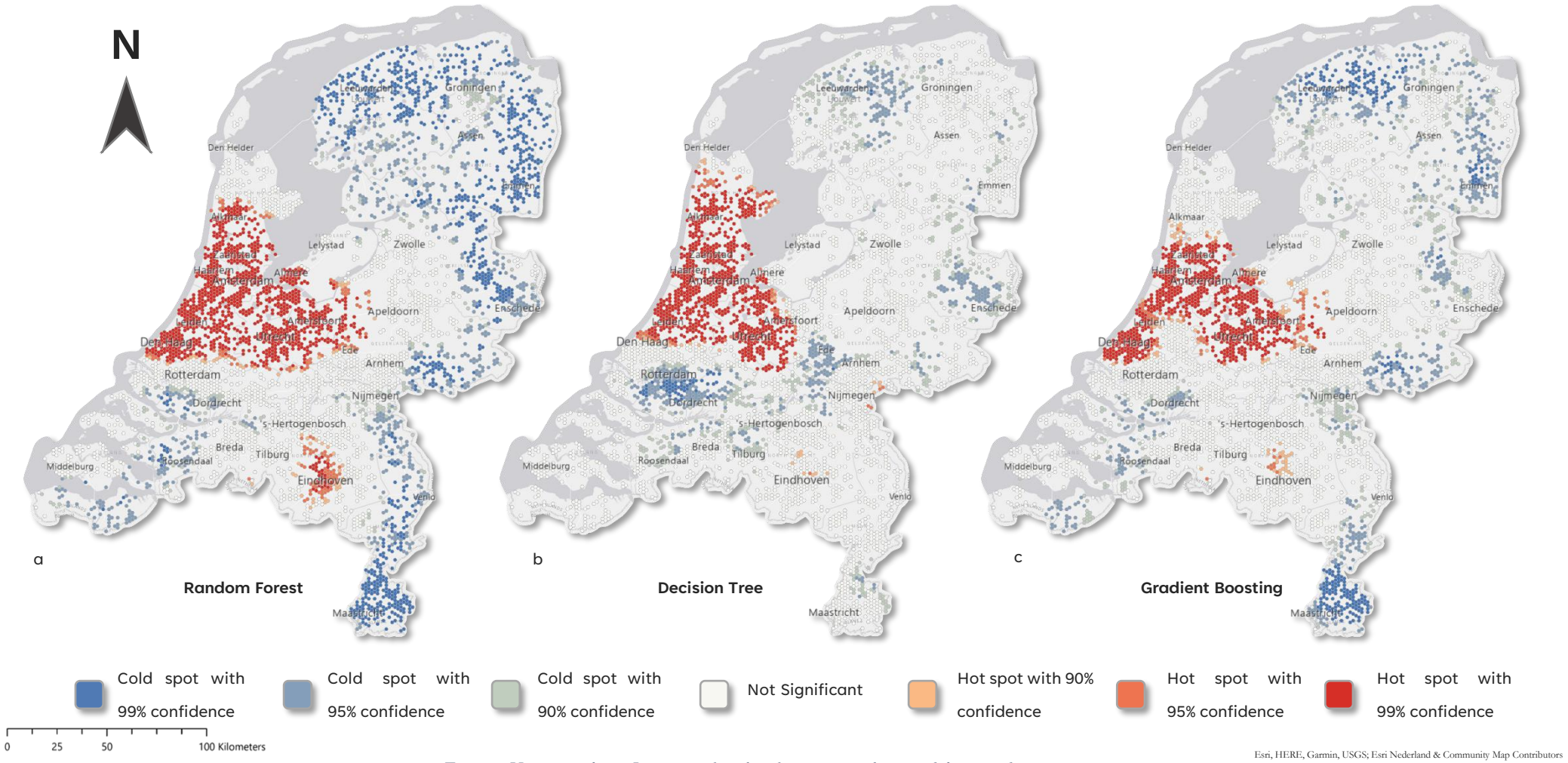



*Figure 29 Hot spot analysis – Importance of number of restaurants within a one kilometre radius*

The maps above show similar patterns as the maps on population density. Again there is a major hot spot in the greater Amsterdam region. Again amongst the big five cities, Rotterdam is the odd one out, showing less importance of this feature.

## Hot spot analysis – Importance of the number of supermarkets within a one kilometre radius

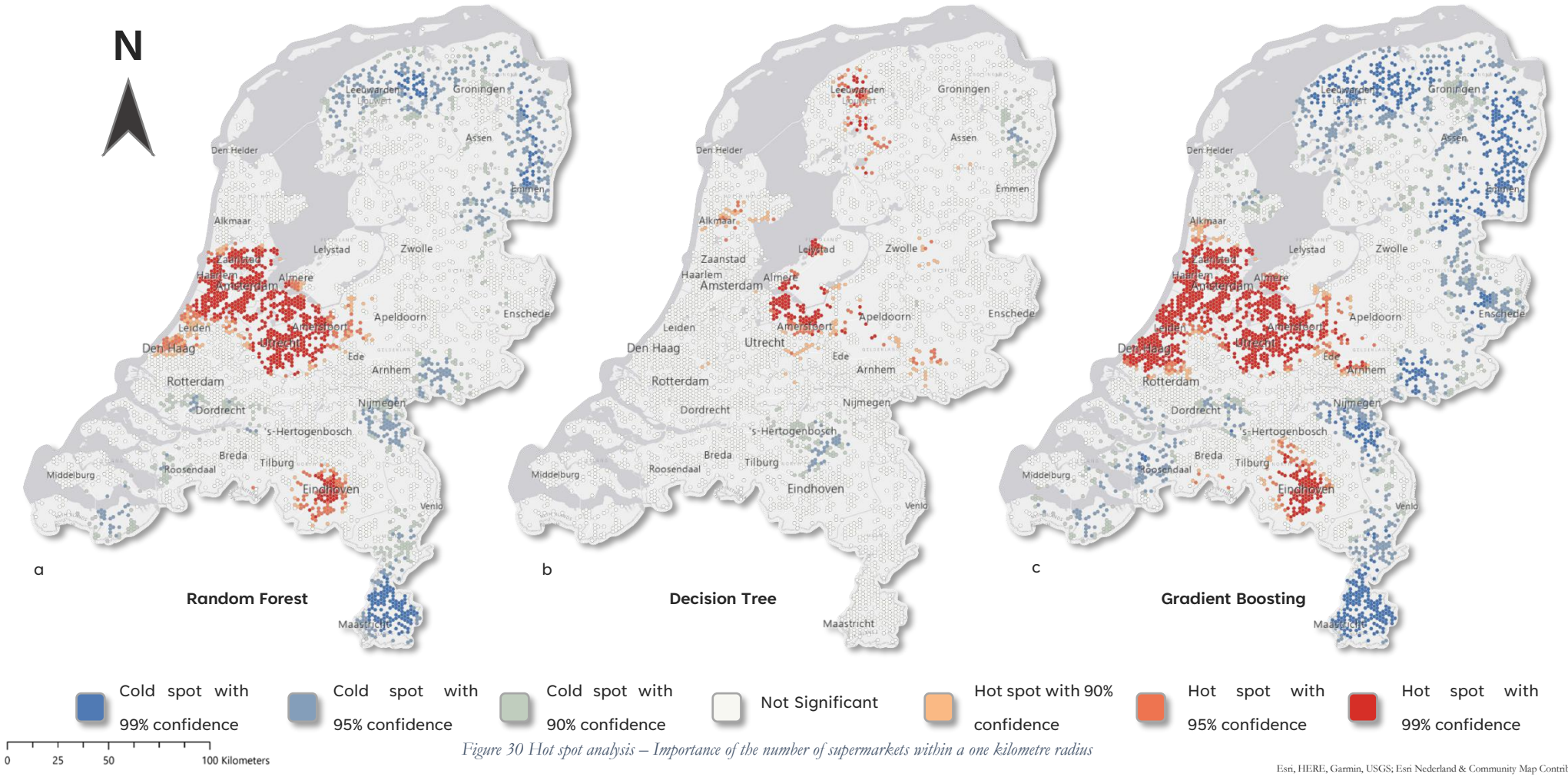



*Figure 30 Hot spot analysis – Importance of the number of supermarkets within a one kilometre radius*

The maps above show similar patterns as the maps on restaurant density. Again there is a major hot spot in the greater Amsterdam region for the RF and the GB algorithm. Again amongst the big five cities, Rotterdam is the odd one out, showing less importance of this feature. The DT algorithm does not show many hot nor cold spots on this feature. This is also in line with the outcomes of the feature importance graph in Figure 18. The DT method shows less importance for both the number of supermarkets and the number of restaurants within a one kilometre radius.

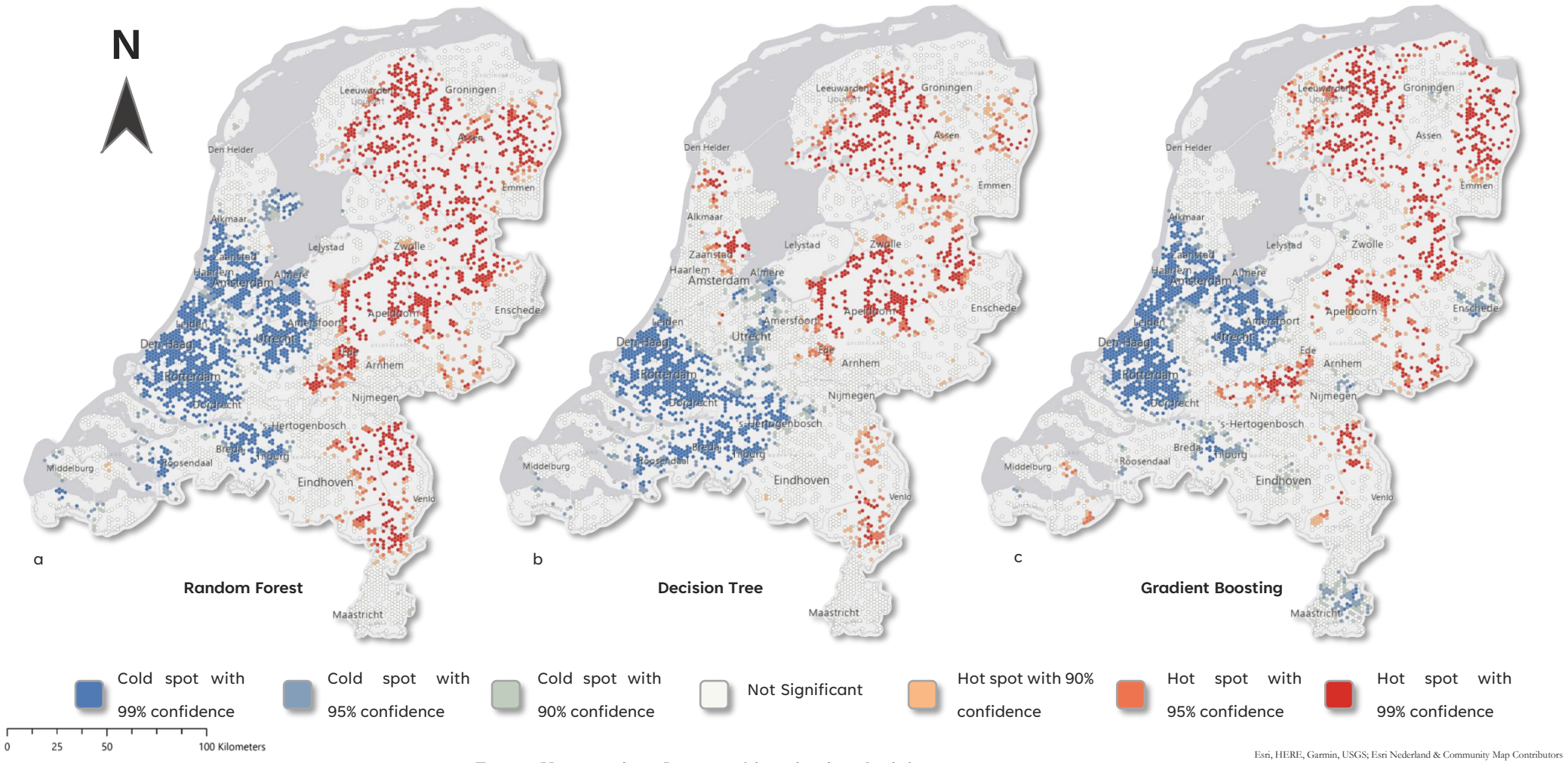



*Figure 31 Hot spot analysis – Importance of the residence being detached*

The maps above show another interesting pattern. The maps portray the importance of the house being detached. Once again a central pattern in the Randstad is visible. The maps show that roughly spoken, the importance of a house being detached is relatively more important to the price prediction for locations outside the Randstad than in the Randstad.

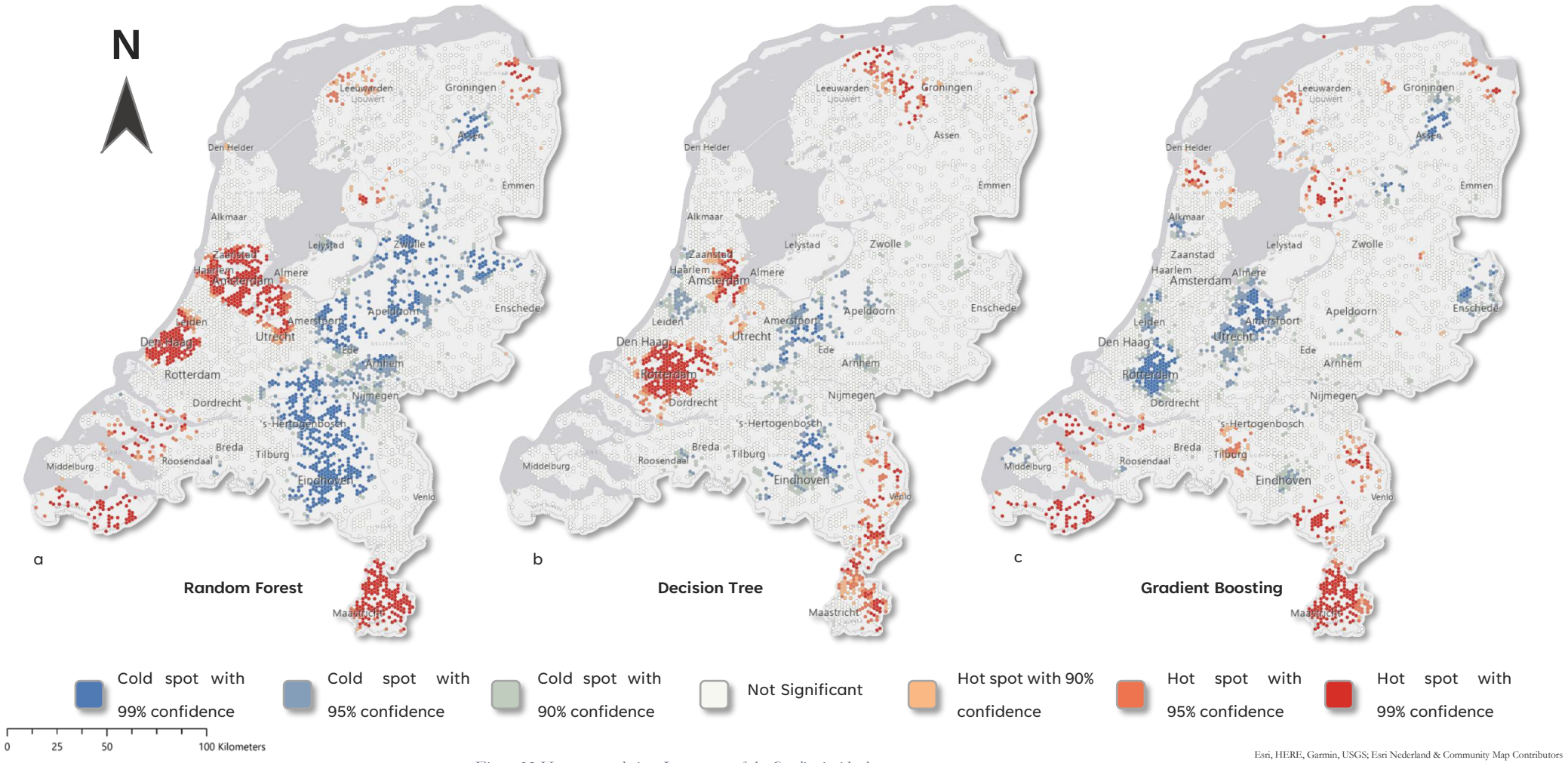



*Figure 32 Hot spot analysis – Importance of the Quality inside the property*

The maps with regards to the quality inside the property differ incredibly. We can observe a similar pattern on the south of the province of Limburg. The most important observation is that the importances per location differ a lot over the three maps.

## Hot spot analysis – Importance of the percentage of western immigrants

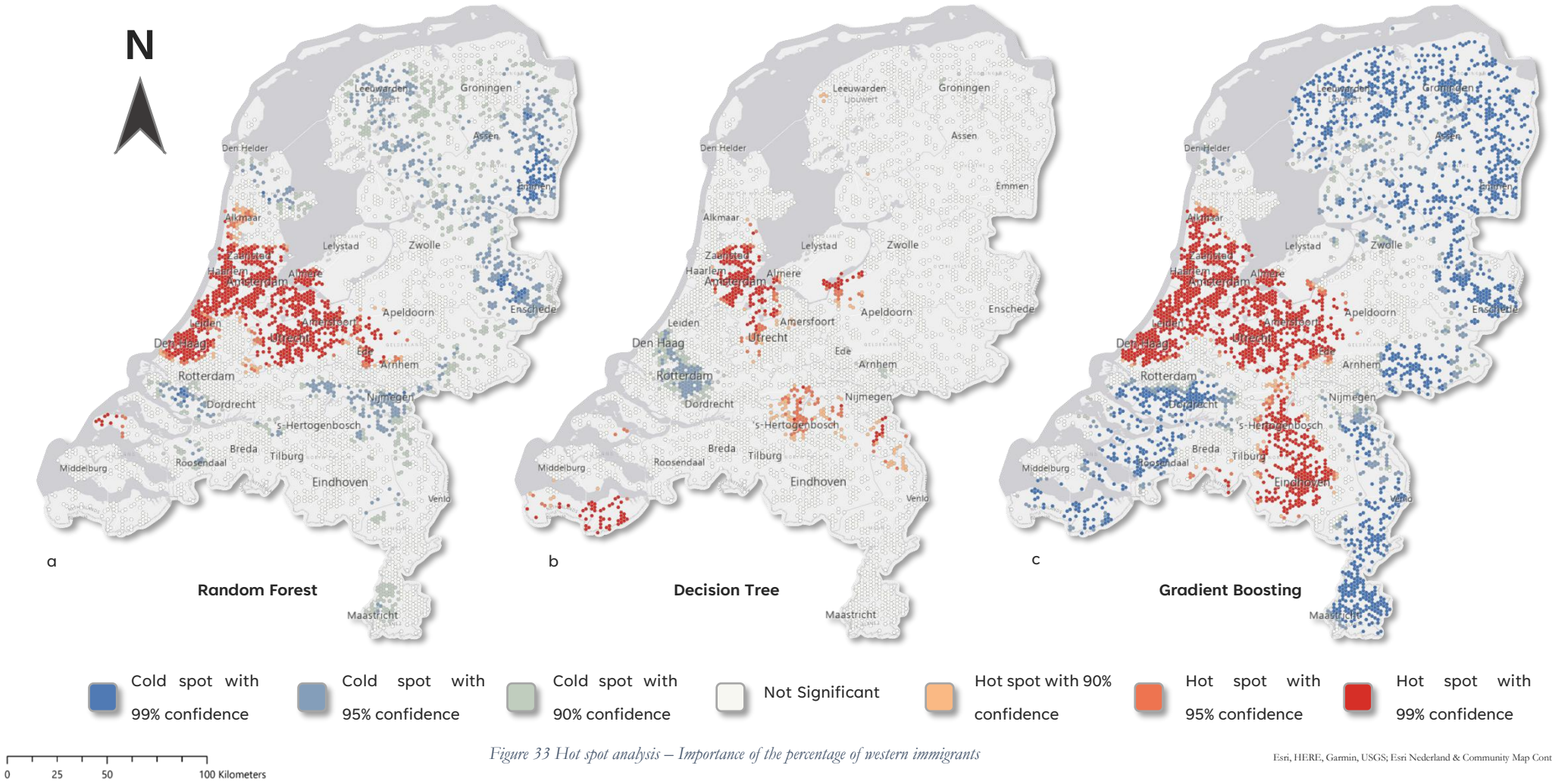



*Figure 33 Hot spot analysis – Importance of the percentage of western immigrants*

The variable percentage of western immigrants in a neighbourhood is more important in and around the cities of The Hague, Utrecht and Amsterdam according to the RF and GB algorithms. The DT method shows a limited number of hotspots compared to the two counterpart. The southern part of Rotterdam shows up as a significant cold spot when it comes to the importance of the percentage of western immigrants.

## 4.4 Spatial regression analysis

To see what features influence the prediction error an OLS regression as well as a spatial error regression were performed. This section presents the outputs of this endeavour. Preceding the regression analysis the correlation matrix in Figure 34 was constructed. This matrix shows some highly correlated independent variables. To prevent a high degree of multicollinearity in the regression models, the selection of the variables were excluded from the regression. The matrix shows some high Pearson R values in the neighbourhood variables. This Figure also shows us preliminary results in the correlation of multiple variables on the prediction error for the three algorithms. We can perceive that show similar directionality in all three algorithms.

The regression outputs show highly significant results in all variables. Two exceptions are the spread and the X coordinate. With regards to the added spatial variable, we can see that it is also significant and is showing a positive correlation with the dependent variable. Furthermore we can observe a higher $R^2$ in all spatial error models. Additionally, the log likelihood has increased in all spatial models compared to the OLS. On top of this we see a lower number for both the Akaike info criterion and the Schwarz criterion. The Breusch-Pagan and Koenker-Basset tests both show significant results, indicating heteroskedasticity. The likelihood ratio test in all spatial error models also show a significant probability. The summary results of the OLS regression and spatial regression are added to the annex (annex I-K).

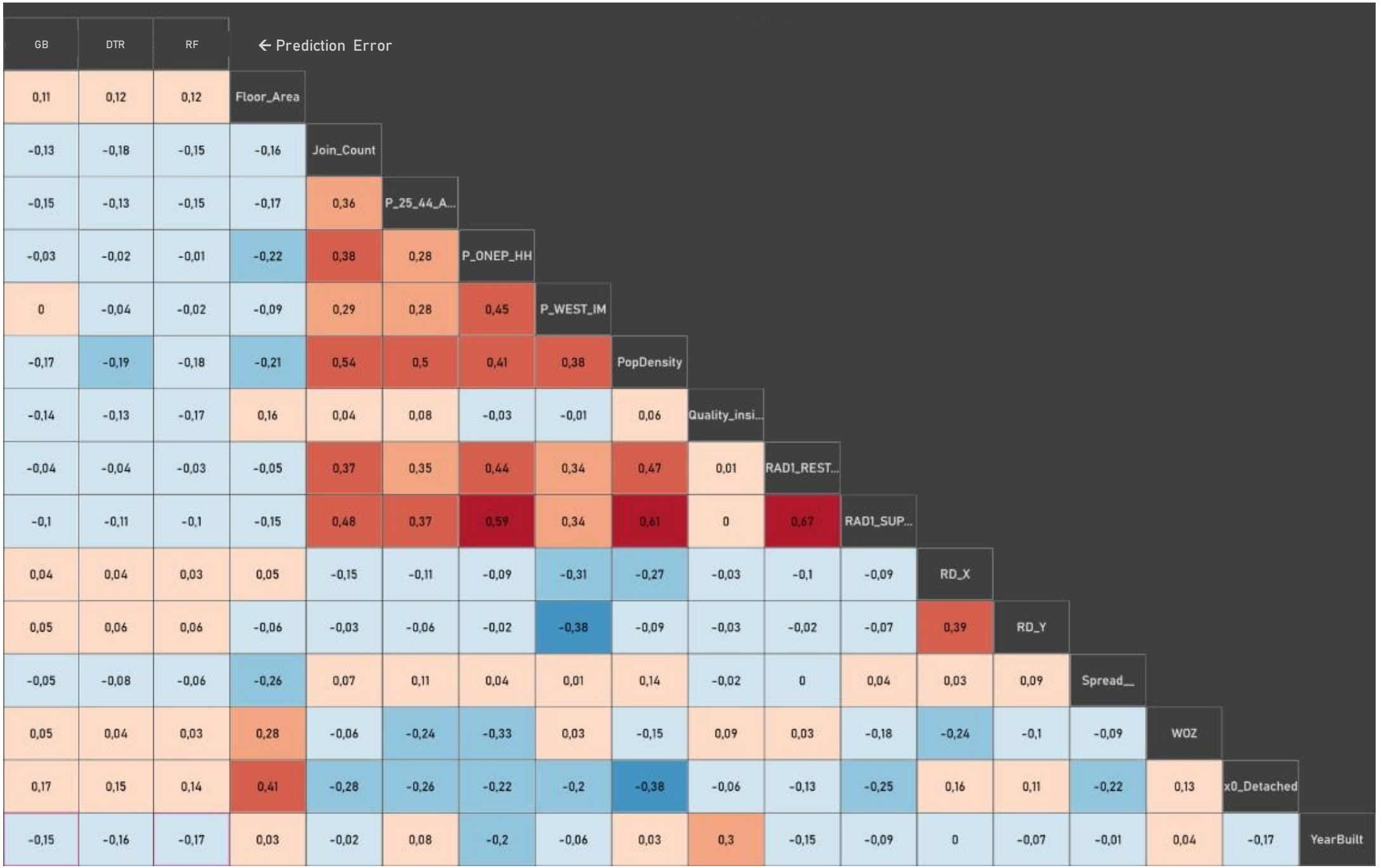



*Figure 34 correlation matrix for the most important variables*

# 5

# Discussion

This chapter discusses limitations and potential biases of the study as well as suggestions for future research. The discussion starts with analysing the performance outcomes from chapter 5 (section 6.1.1). Following that the results from the spatial prediction accuracy are assessed (section 6.1.2). Section 6.1.3 discusses the outcomes of the maps presented in section 5.3 after which the results from the spatial regression are critically reflected on (section 6.1.4). Sections 6.2 and 6.3 discuss the limitations of this study and provides starting points for future research respectively.

## 5.1.1 Discussing performance outcomes

From the performance metrics in Figure 13 we can see that the Random Forest algorithm outperforms both the Gradient Boosting algorithm and the Decision Tree algorithm. The mean absolute error for the Random Forest shows a deviation of €387,77, indicating that the predictions are on average 11,2% off. When controlling for outliers this number drops to 8,3%. Comparing this result to the assessment of uniformity stipulated in the Standard on Ratio studies shows that the outcomes fall within the acceptable range of mass appraisals for residential real estate (International Association of Assessing Officers, 2013). The fact that the Random Forest algorithm outperforms the Gradient Boosting algorithm is in contrast with the findings in Buodd & Derås (2020). However, this study includes the hyperparameter tuning of the algorithms to fit the data best.

The number of accurate, overestimated and underestimated observations are counted as their aggregated median and do not depict the total number of predicted transactions. Having these outcomes aggregated means on the one hand that we can say with more certainty that cells contain over or underpredictions and attribute this to a pattern rather than random error. On the other hand it means that we're ignoring individual transactions that might have been of importance for showing emerging patterns. The map in Figure 14 shows that the number of underprediction is highest in the Gradient Boosting algorithm as compared to

the other algorithms. It seems as though there is no scientific evidence that backs this claim yet. On the contrary, the Decision Tree algorithm is most likely to overestimate. All algorithms show signs that underprediction is more prevalent than overprediction in these algorithms. What is interesting is that the Decision Tree algorithm performs slightly higher than the Gradient Boosting algorithm when it comes to simply counting the accurate predictions within a 5% range. However, the metrics show that the deviation is the largest in this algorithm. It is surprising that the variance is highest in the decision tree, but competitive with the Gradient Boosting.

### 5.1.2 Discussing spatial patterns in prediction accuracy

Section 2 of chapter 7 shows that using this dataset hot spots of the percentage error cluster mostly around the central cities in The Netherlands. These maps indicate that these places are the ones where overestimation occurs most. This is an interesting observation as research states that the housing prices in these regions are in reality also overinflated (Hochstenbach & Arundel, 2019) (Figure 14).

**Figure 3. Neighbourhood-level percentage change of house values for the 2006-2018 period (left map) and the 2015-2018 period (right map). The cartogram is distorted based on the number of dwellings per neighbourhood. Source: SSD, own calculations.**

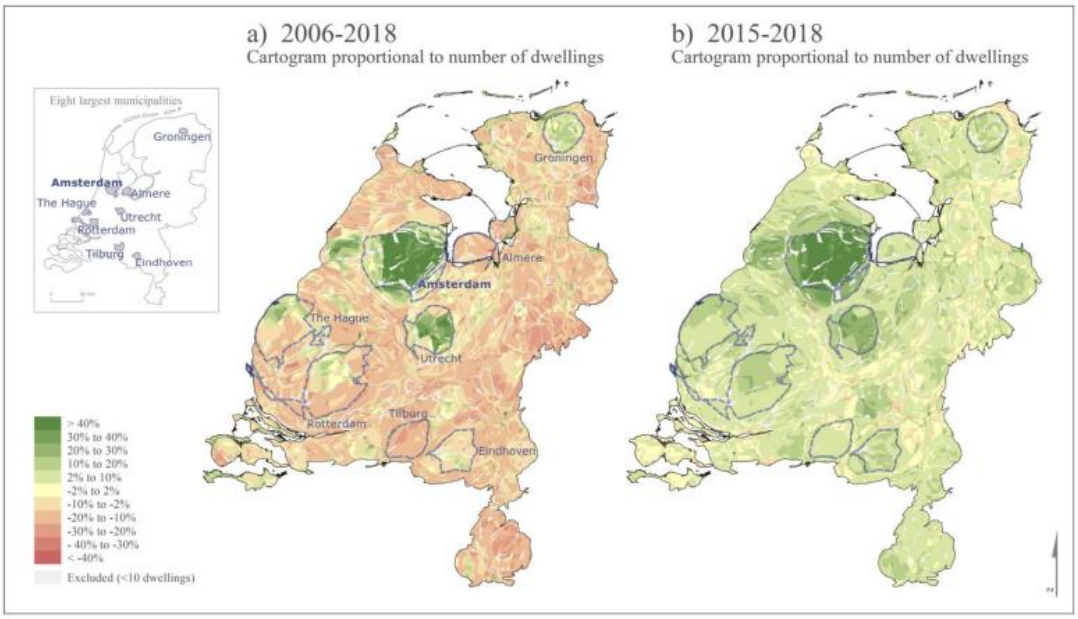



*Figure 35 Neighbourhood level percentage change (Hochstenbach & Arundel, 2019)*

When analysing the absolute prediction error, significant cold spots emerge around the city of Rotterdam and Eindhoven, indicating that the prediction accuracy is highest in these regions. Further research is necessary to explain why these regions perform higher than the rest.

### 5.1.3 Discussing emerging patters of feature importance

The presence of multicollinearity from the regression analysis and correlation matrix is also visible when cross comparing the maps from different variables maps. For example, tax based values and construction year show similar patterns in terms of hot and cold spots. This might be an indication that these factors are

related in terms of feature importance. The maps show that most of the times the hotspots occur in and around the Randstad, which is interesting to see on its own. However, the importance of the residence being detached shows cold spots in this region. This might be attributed to the fact that there are relatively few detached dwellings in this region. It also shows that in the regions where hotspots occur, according to this variable, having a detached house plays a bigger role in predicting the price as compared to other types of dwellings. This might show that people value a free standing house more in less populated areas.

While most of the times that maps from different algorithms on the same variable are more or less in accordance with each other, some conflicting maps do stand out. One such example is the map on the quality inside of a property. This map shows that the Random Forest and decision tree methods show similar results, with hotspots in Amsterdam, Maastricht and between The Hague and Rotterdam. However, the Gradient Boosting map shows colds pots in the area of Amsterdam and Rotterdam. These patterns stand out and might need further investigating to draw a conclusion on why this happens.

An additional observation can be made when comparing the importance of the quality inside the house and the importance of a dwelling being detached. The maps show some signs of inversely correlated relationships between those features. This might indicate that where the importance of a house being detached, the importance is of the quality is of not much relevance. The importance of the spread, or the difference between the asking price and the price paid, is highest in the areas of Utrecht, Amsterdam, Nijmegen and Groningen. These regions are seen as fiercely competitive markets where asking prices tend to be overbid by some margin. The importance of this feature in this region shows that the overbidding plays an significant role when determining the eventual market price in these regions.

### 5.1.4 Discussing the correlation with prediction error

The study has found that for this dataset it shows that features don't have the same importance in all geographic regions. The linear regression models show that there is correlation between the prediction error and the independent variables. The study has indicated that the spatial error models perform slightly better than the OLS model. The increased log likelihood of the spatial error model indicates that the addition of a spatial variable increases the model fit. The Koenker-Bassett test and Bresuch-Pagan test indicate that the residuals are not normally distributed. This might indicate that local differences are too complex to model in a linear regression. The outputs of the regression are therefore difficult to interpret as the residuals are not normally distributed. However, we do see that the number of transactions in an area, the usable floor area and the fact that a house is detached play an important role in the correlation with the prediction error.

## 5.2 Limitations

The algorithms used for this research have not been parametrically tuned to perfection. This implies that we have to consider the fact that the outcomes of these models might differ when the models are sufficiently tweaked by changing the parameters it runs on. Moreover, the algorithms were merely ran once. Ideally the

model would have to be run several times and their outputs be averaged to come to a more definitive conclusion on performance as has been previously done (Ho et al., 2021; Yilmazer & Kocaman, 2020).
In hindsight the feature permutation importance would have been a better metric to use as feature importance. The study has shown that the categorical variables haven't shown much importance in the feature importance calculation. This can be observed from the number of categorical variables that occur in the list of top features. These findings are in line with the findings in other studies where this methods were compared (Scikit Learn, 2023; Strobl et al., 2007). Future research might be better of using the feature permutation importance as categorical variables (eg. Energy label) are not to be ignored (Brainbay, 2022). The fact that building status did not show up as an important factor can be attributed to the number of new-built entries.

While this study was limited to a dataset covering only transactions in 2021 it is easily replicable and expandable with bigger datasets. The UML model, in combination with the python scripts will make an extension of this research relatively straightforward.

Additionally, temporal trends have not been analysed due to the fact that data was limited for this research. A higher model fit might be achieved by incorporating datapoints from multiple years to show the impact on the prediction error.

# 6

# Conclusion

## 6.1 Spatial Differences in the Housing Market

This chapter summarizes the findings and outcomes of the research done to investigate if the prediction accuracy of different machine learning algorithms are subjected to spatial patterns. The research posed to answer the question of how spatial statistics can help us better understand the outcomes of machine learning AVM's.

What has been shown with this study is that not all housing markets are created equally. The prediction accuracy of the Random Forest, Gradient Boosting and Decision Tree algorithms were measured using several indicators. Additionally, the parameters of the algorithms were not tuned to be able to set a baseline for further research. All metrics indicate that the Random Forest algorithm performs best on this dataset showing high performance, especially compared to the non-ensemble method counterpart.

The first sub question was aimed towards finding suitable metrics to measure the accuracy of the predictions for the model as well as the individual transactions. The accuracy of the individual transactions were measured using the absolute percentage error and the percentage error. Using this calculation we can conclude that the Random Forest also has the most accurate observations within a 5% error range. Moreover, we can see that the Decision Tree algorithm tends to overpredict most, and that the Gradient Boosting algorithm has the tendency to underpredict most.

The second sub question was set to understand spatial patterns in the prediction error. Reading the outputs of the maps presented we can conclude that emerging cold spot patterns in the regions of Rotterdam, Eindhoven and Zwolle might indicate that the prediction accuracy is highest in these areas (absolute percentage error). In other words, the models generated, perform best in these regions. These regions are similar to the hotspots that can be observed in the maps that show the percentage error, indicating that locations with high accuracy tend to be more overestimated than underestimated. Additionally, the places mentioned are locations with a relatively high number of transactions. However, cities like Utrecht and Amsterdam don't show similar patterns of accuracy, which might indicate that there is correlation with other variables. With regards to the global spatial autocorrelation, the study shows that

the Random Forest algorithm shows a slightly higher Moran's I compared to the other algorithms. This can possibly show that the Random Forest algorithm weighs nearby factors more than the Gradient Boosting or Decision Tree algorithms.

The third sub question posed to answer the question of whether certain features are more prevalent in some places than in others. Determining feature importance makes up a central part of this question. After analysing the sum of the importance of all features, the study shows that neighbourhood, spatial and structural variables contribute to about 94% of the total feature importance. Spatial variables, containing the X and Y coordinate as well as the grid cell number, show the highest importance to number of variables. This may show signs of importance of spatiality in determining housing prices using an AVM. The maps on pages 44 through 57 show hot and cold spots of feature importance on the most important variables. These maps show the differences and similarities between the algorithms in terms of local spatial autocorrelation. In general, the three algorithms show many similar patterns in feature importance. However, some notable exceptions can be found when carefully reading the maps on useable floor area, the construction year and the supermarkets within a one kilometre radius. The results also show comparable results throughout different variables, which might indicate correlation of these variables. One such observation can be made between the variables average tax based value in a neighbourhood and the construction year.

The final sub-question answers which variables significantly affect the prediction error. After careful examination of the most important variables and their spatial patterns the study draws a conclusion on which factors impact the prediction accuracy. Results of the spatial regression show that the floor area, building year, number of restaurants, percentage of one person households, Y coordinate and the fact that a house is detached are positively related to the prediction error in all three algorithms. All other variables in the top 15 show a negative correlation or are insignificant. Moreover we see an improvement in the spatial error model as compared to the OLS model, meaning that a spatial regression model fits the dataset better.

To conclude we can carefully assume that spatial patterns in the prediction accuracy do occur when using machine learning models to predict the price of a house. We have seen a high baseline performance of the Random Forest algorithm and can be further tuned to produce better results. The study has also shown that there might be slight differences in where features play an important role in this price prediction. Outcomes of this study have shown that the models of the real estate market in The Netherlands in 2021 perform different, in different locations. The study shows that variables play a different role in different places These outcomes might help planners, decision makers and appraisers in the future to better understand what's important in different locales. Acknowledging the differences in geography has shown to be important when assessing properties on a large scale, and will likely grow to be more important.

## 6.2 Future research

This research has focussed on showing the spatial importance of different variables in different locations. The study did not include the spatial analysis of SHAP values. A study on this topic might indicate the

directionality of the variables. Both hot and colds pots might be interesting in this case as we can see where variables are positive and negatively correlated with the prediction price. A study in this case might focus on finding the different types of relations between variables. One could use the function local bivariate relationships for this purpose.

Moreover, future study might analyse why the discovered patterns in feature importance occur. One might for example state that in a locale where demand is high and supply is low, features are oftentimes less important than in places where choice is abundant. This research however is limited to recognizing the patterns visible in the feature importance. Studies in the future might focus more on the magnitude of the importance in relative or absolute terms. Furthermore, an opportunity is opened to look for patterns in different state of the art algorithms like Support Vector Machine Learning and Artificial Neural Networks. In addition to this, research might venture out to synthesise new and important variables to be used in models like these.

This research used OLS and the spatial error model to predict the prediction error. Even though the spatial models seemed to perform better than the OLS regression, there were still signs of heteroskedasticity. Therefore, geographically weighted regression (GWR) local linear correlations might be a suitable alternative for the spatial error model to explain more of the variance on different levels. This study tried to perform a GWR on the dataset, however encountered issues due to the constraint in coverage. This regression was therefore unsuccessful but might be able to perform on larger datasets.

# Bibliography


Aladag, C., Egrioglu, E., & Kadilar, C. (2012). Improvement in Forecasting Accuracy Using the Hybrid Model of ARFIMA and Feed Forward Neural Network. *American Journal of Intelligent Systems*, *2*, 12–17. https://doi.org/10.5923/j.ajis.20120202.02

Anselin, L. (2005). Exploring spatial data with GeoDaTM: a workbook. *Center for Spatially Integrated Social Science*, *1963*, 157.

Beimer, J., & Francke, M. (2019). Out-of-sample house price prediction by hedonic price models and machine learning algorithms. *Real Estate Research Quarterly*, *18*(2), 13–20.

Bell, S., & Owusu, B. (2021). *Spatial Regression in GeoDa*. https://doi.org/10.13140/RG.2.2.25865.98409

Bensdorp, V. R. J. (2021). *Influence of population demographics on real estate prices in Zuid-Holland*. Utrecht University.

Bidanset, P. E., & Lombard, J. R. (2014). A Comparison of Geographically Weighted Regression and the Spatial Lag Model. *Cityscape*, *16*(3), 169–182. http://www.jstor.org.proxy.library.uu.nl/stable/26326913

Brainbay. (2022). *Uitdraai woontransacties 2021 (accessed by Capital Value on 1-11-2022) [Unpublished Dataset]*.

Brainbay. (2022). *De waarde van het energielabel – investeren in duurzaamheid loont steeds meer*. https://www.nvm.nl/nieuws/2022/de-waarde-van-het-energielabel/

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Buodd, M. F., & Derås, E. J. (2020). Machine learning for property valuation : an empirical study of how property price predictions can improve property tax estimations in Norway. *Master Thesis*.

Cagdas, V., Kara, A., van Oosterom, P., Lemmen, C., Işıkdağ, Ü., Kathmann, R., & Stubkjær, E. (2016). An initial design of ISO 19152: 2012 LADM based valuation and taxation data model. *Isprs Journal of Photogrammetry and Remote Sensing*, *4*, 145–154.

CBS. (2021). *Kerncijfers wijken en buurten 2021 [Dataset]*. https://www.cbs.nl/nl-nl/dossier/nederland-regionaal/geografische-data/wijk-en-buurtkaart-2021

Chaphalkar, Dr. N. B., & Sandbhor, S. (2013). Use of Artificial Intelligence in Real Property Valuation. *International Journal of Engineering and Technology (IJET)*

Chen, X., Ye, X., Widener, M. J., Delmelle, E., Kwan, M.-P., Shannon, J., Racine, E. F., Adams, A., Liang, L., & Jia, P. (2022). A systematic review of the modifiable areal unit problem (MAUP) in community 1

Chou, J.-S., & Bui, D.-K. (2014). Modeling heating and cooling loads by artificial intelligence for energy-efficient building design. *Energy and Buildings*, *82*, 437–446. https://doi.org/10.1016/j.enbuild.2014.07.036

Chou, J.-S., Fleshman, D.-B., & Truong, D.-N. (2022). Comparison of machine learning models to provide preliminary forecasts of real estate prices. *Journal of Housing and the Built Environment*, *37*(4), 2079–2114. https://doi.org/10.1007/s10901-022-09937-1

Christie, D., & Neill, S. P. (2022). 8.09 - Measuring and Observing the Ocean Renewable Energy Resource. In T. M. Letcher (Ed.), *Comprehensive Renewable Energy (Second Edition)* (pp. 149–175). Elsevier. https://doi.org/10.1016/B978-0-12-819727-1.00083-2

Davis, P., McCord, M., Bidanset, P., & Cusack, M. (2020). *Nationwide Mass Appraisal Modeling in China: Feasibility Analysis for Scalability Given Ad Valorem Property Tax Reform*.

Frye, C., de Mijolla, D., Begley, T., Cowton, L., Stanley, M., & Feige, I. (2020). Shapley explainability on the data manifold. *ArXiv Preprint ArXiv:2006.01272*.

Guliker, E., Folmer, E., & van Sinderen, M. (2022). Spatial Determinants of Real Estate Appraisals in The Netherlands: A Machine Learning Approach. *ISPRS International Journal of Geo-Information*, *11*(2), 125.

Haining, R. P. (2001). Spatial Autocorrelation. In N. J. Smelser & P. B. Baltes (Eds.), *International Encyclopedia of the Social & Behavioral Sciences* (pp. 14763–14768). Pergamon. https://doi.org/10.1016/B0-08-043076-7/02511-0

Helbich, M., Brunauer, W., Vaz, E., & Nijkamp, P. (2013). Spatial Heterogeneity in Hedonic House Price Models: The Case of Austria. *Urban Studies*, *51*(2), 390–411. https://doi.org/10.1177/0042098013492234

Ho, W. K. O., Tang, B.-S., & Wong, S. W. (2021). Predicting property prices with machine learning algorithms. *Journal of Property Research*, *38*(1), 48–70. https://doi.org/10.1080/09599916.2020.1832558

Hoang, D., & Wiegratz, K. (2022). Machine learning methods in finance: Recent applications and prospects. *European Financial Management*, *n/a*(n/a). https://doi.org/https://doi.org/10.1111/eufm.12408

Hochstenbach, C., & Arundel, R. (2019). *Spatial housing-market polarization: diverging house values in the Netherlands*. CUS Working Paper Series-WPS.

Hodson, T. O., Over, T. M., & Foks, S. S. (2021). Mean Squared Error, Deconstructed. *Journal of Advances in Modeling Earth Systems*, *13*(12), e2021MS002681. https://doi.org/https://doi.org/10.1029/2021MS002681

Hu, L., Chun, Y., & Griffith, D. A. (2022). Incorporating spatial autocorrelation into house sale price prediction using random forest model. *Transactions in GIS*, *26*(5), 2123–2144. https://doi.org/https://doi.org/10.1111/tgis.12931

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, *22*(4), 679–688. https://doi.org/10.1016/j.ijforecast.2006.03.001

International Association of Assessing Officers. (2013). *Standard on Ratio Studies*.

International Association of Assessing Officers. (2016). *Standard on Verification and Adjustment of Sales*.

International Association of Assessing Officers. (2018). *Standard on Automated Valuation Models (AVMs)*.

Joseph, V. R. (2022). Optimal ratio for data splitting. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *15*(4), 531–538. https://doi.org/10.1002/sam.11583

Kars, J. (2021). *Predicting Neighborhood Prices: Machine Learning and Hedonic Pricing In The Dutch Housing Market*. Tilburg University.

Kim, S., & Kim, H. (2016). A new metric of absolute percentage error for intermittent demand forecasts. *International Journal of Forecasting*, *32*(3), 669–679. https://doi.org/10.1016/j.ijforecast.2015.12.003

Krause, A., & Lipscomb, C. (2016). The Data Preparation Process in Real Estate: Guidance and Review. *Journal of Real Estate Practice and Education*, *In Press*. https://doi.org/10.1080/10835547.2016.12091756

Lemmen, C., da SILVA, A., & Chipofya, M. (2022). LADM in the Classroom–Making the Land Administration Domain Model Accessible. *27th FIG Congress 2022: Volunteering for the Future-Geospatial Excellence for a Better Living*.

Lubberink, A. S., van der Post, W., & Veuger, J. (2017). De WOZ-waarde als marktwaarde-indicator. *Real Estate Research Quarterly*, *16*(4), 28–37.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, *30*.

Mueller, J. P., & Massaron, L. (2021). *Machine learning for dummies*. John Wiley & Sons.

Ng, A., & Deisenroth, M. (2015). Machine learning for a London housing price prediction mobile application. *Imperial College London*.

Ngiam, K. Y., & Khor, I. W. (2019). Big data and machine learning algorithms for health-care delivery. *The Lancet Oncology*, *20*(5), e262–e273. https://doi.org/10.1016/S1470-2045(19)30149-4

NVM. (2022). *De Waarde van het Energielabel - investeren in duurzaamheid loont steeds meer*.

Park, B., & Bae, J. K. (2015). Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data. *Expert Systems with Applications*, *42*(6), 2928–2934. https://doi.org/10.1016/j.eswa.2014.11.040

Peterson, S., & Flanagan, A. (2009). Neural network hedonic pricing models in mass real estate appraisal. *Journal of Real Estate Research*, *31*(2), 147–164.

Phan, T. D. (2018). Housing Price Prediction Using Machine Learning Algorithms: The Case of Melbourne City, Australia. *2018 International Conference on Machine Learning and Data Engineering (ICMLDE)*, 35–42. https://doi.org/10.1109/iCMLDE.2018.00017

Plaia, A., Buscemi, S., Fürnkranz, J., & Mencía, E. L. (2022). Comparing Boosting and Bagging for Decision Trees of Rankings. *Journal of Classification*, *39*(1), 78–99. https://doi.org/10.1007/s00357-021-09397-2

Schiewe, J. (2021). Distortion Effects in Equal Area Unit Maps. *KN - Journal of Cartography and Geographic Information*, *71*(2), 71–82. https://doi.org/10.1007/s42489-021-00072-5

Schneider, P., & Xhafa, F. (2022). Chapter 3 - Anomaly detection: Concepts and methods. In P. Schneider & F. Xhafa (Eds.), *Anomaly Detection and Complex Event Processing over IoT Data Streams* (pp. 49–66). Academic Press. https://doi.org/10.1016/B978-0-12-823818-9.00013-4

Scikit learn. (2022). *Scikit-Learn Machine Learning in Python*. https://scikit-learn.org/stable/

Scikit Learn. (2023). *Permutation feature importance*.

Strobl, C., Boulesteix, A.-L., Zeileis, A., & Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, *8*(1), 25. https://doi.org/10.1186/1471-2105-8-25

Verbeek, T. (2018). Unequal Residential Exposure to Air Pollution and Noise: A Geospatial Environmental Justice Analysis for Ghent, Belgium. *SSM - Population Health*, *7*, 100340. https://doi.org/10.1016/j.ssmph.2018.100340

Waarderingskamer. (2022, October 27). *Onderzoeken kwaliteit taxaties (inclusief ratio studies)*.

Yilmazer, S., & Kocaman, S. (2020). A mass appraisal assessment study using machine learning based on multiple regression and random forest. *Land Use Policy*, *99*, 104889. https://doi.org/10.1016/j.landusepol.2020.104889

# Appendix

## A. Script for Random Forest algorithm

```
import numpy as np
import pandas as pd
import shap
import math
import matplotlib.pyplot as plt
from shap.plots import _waterfall
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_percentage_error
from sklearn.metrics import r2_score

# load the prepared datafile
filepath = 'Z:/Private/f.berks/Thesis/Data/Dataset_CleanedV3.csv'
dataset = pd.read_csv(filepath)

# State name for saving
version = "2.0"
model_name = "Random_Forest"
file_name = model_name+"_"+version
print(file_name)

# dataset sample for faster tweaking and running
dataset = dataset.sample(100)

# X (Independent variables) and y (target variable)
X = dataset.drop(columns={'Transactionprice_m2'}, axis=1)
y = dataset['Transactionprice_m2']

# create headers for the Shap values outcomes
headers_list = list(X.columns)
headers_list_shap = ['shap_'+ item for item in headers_list]


#Splitting the data into train,test data
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=
0)

# define the model
model = RandomForestRegressor(n_estimators=100,random_state=0)
# train the model
model.fit(X_train,y_train)

# run the model
y_pred = model.predict(X_test)
```

```
# Create Tree Explainer object that can calculate shap values
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X_test,approximate=True)
#get the feature importance values from the model
importances = model.feature_importances_

#create a dataframe with the feature names and importance values
feature_importance = pd.DataFrame({'feature': X.columns, 'RFimportance':
importances})

#concatenate the two tables with the predicted and actual values.
X_test['actual value'] = y_test
X_test['predicted value'] = y_pred
X_test['prediction error'] = abs(y_test-y_pred)/y_test
X_test[headers_list_shap] = shap_values
results = X_test

# run statistical scores to assess the performance of the model
mse = mean_squared_error(y_test, y_pred)
rmse = math.sqrt(mean_squared_error(y_test, y_pred))
stderr = rmse / math.sqrt(len(y))
mean_average_error = sum(abs(y_pred - y_test)) / len(y_test)
mape = mean_absolute_percentage_error(y_test, y_pred)
mdape = np.median((np.abs(np.subtract(y_test, y_pred)/ y_test))) * 100
R2 = r2_score(y_test, y_pred)
#create dataframe for the statistical scores
scores = {'score
name':['mse','rmse','stderr','mean_average_error','mape','mdape","R2"],'Value'
:[mse,rmse,stderr,mean_average_error,mape,mdape,R2]}
df_scores = pd.DataFrame(scores)
print(df_scores)

# save documents to CSV format
test_filepath = 'Z:/Private/f.berks/Thesis/Data/tests/'
results_filepath = str(test_filepath + model_name + "V" + version +
'_results'+'.csv')
results.to_csv(results_filepath, sep=',')
scores_filepath = str(test_filepath + model_name + "V" + version +
'_scores'+'.csv')
df_scores.to_csv(scores_filepath, sep=',')
feature_importance_filepath = str(test_filepath + model_name + "V" + version +
"_featureImportance" +'.csv')
feature_importance.to_csv(feature_importance_filepath, sep=',')
```

## B. Script for Gradient Boosting algorithm

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import shap
import math

from sklearn.model_selection import train_test_split
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_percentage_error
from sklearn.metrics import r2_score

# load the prepared datafile
filepath = 'Z:/Private/f.berks/Thesis/Data/Dataset_CleanedV3.csv'
dataset = pd.read_csv(filepath)

# State name for saving
version = "1.0"
model_name = "Gradient_Boosting"
file_name = model_name+"_"+version
print(file_name)

# dataset sample for faster tweaking and running
dataset = dataset.sample(100)

# X (Independent variables) and y (target variable)
X = dataset.drop(columns={'Transactionprice_m2'}, axis=1)
y = dataset['Transactionprice_m2']

# create headers for the Shap values outcomes
headers_list = list(X.columns)
headers_list_shap = ['shap_'+ item for item in headers_list]

#Splitting the data into train,test data
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=
0)


model = GradientBoostingRegressor(n_estimators=100,random_state=0)
model.fit(X_train,y_train)
y_pred = model.predict(X_test)

# Create Tree Explainer object that can calculate shap values
explainer = shap.TreeExplainer(model)
```

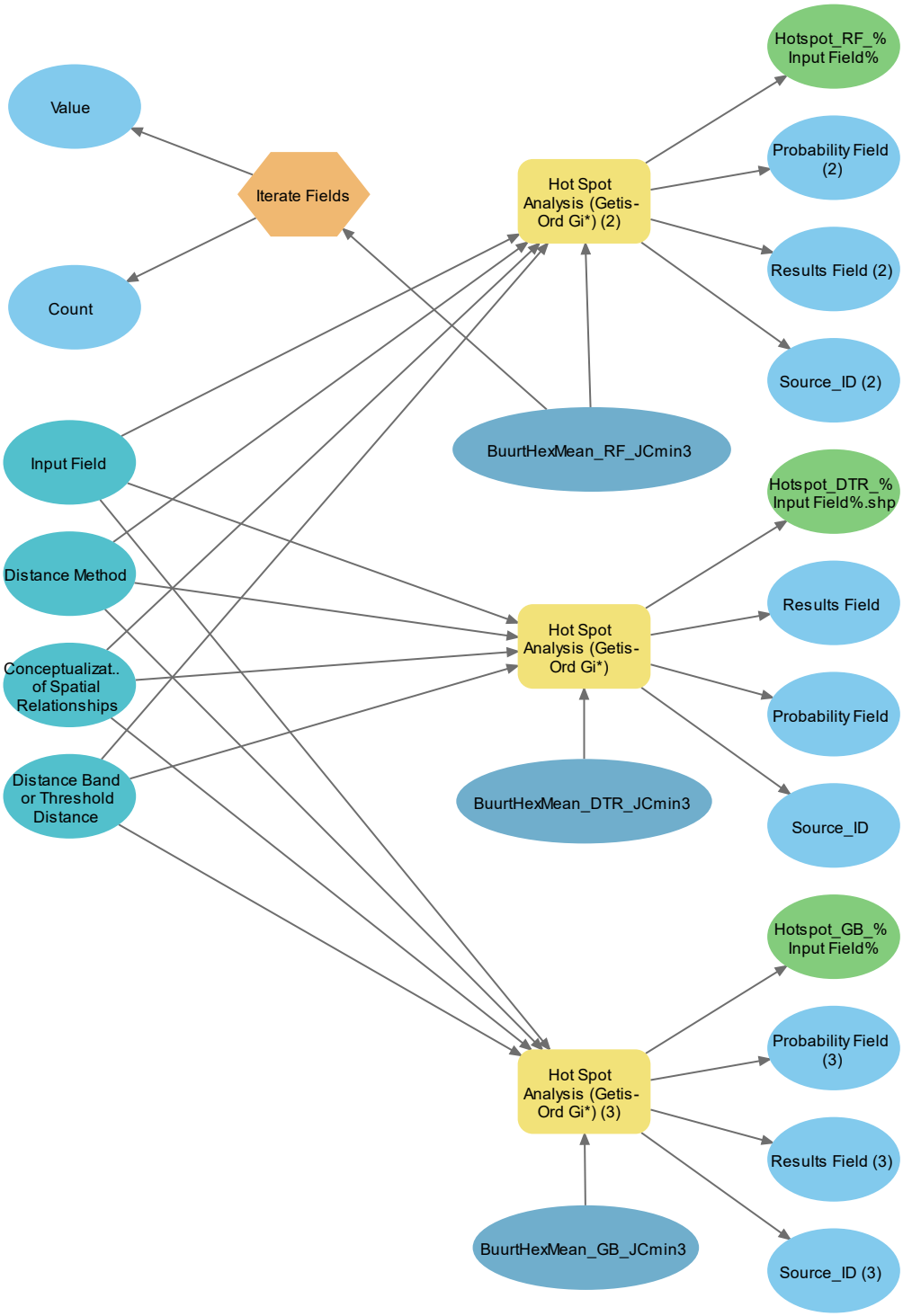
```
shap_values = explainer.shap_values(X_test,approximate=True)
#get the feature importance values from the model
importances = model.feature_importances_

#create a dataframe with the feature names and importance values
feature_importance = pd.DataFrame({'feature': X.columns, 'GBimportance':
importances})

#concatenate the two tables with the predicted and actual values.
X_test['actual value'] = y_test
X_test['predicted value'] = y_pred
X_test['prediction error'] = abs(y_test-y_pred)/y_test
X_test[headers_list_shap] = shap_values

results = X_test

# run statistical scores to assess the performance of the model
mse = mean_squared_error(y_test, y_pred)
rmse = math.sqrt(mean_squared_error(y_test, y_pred))
stderr = rmse / math.sqrt(len(y))
mean_average_error = sum(abs(y_pred - y_test)) / len(y_test)
mape = mean_absolute_percentage_error(y_test, y_pred)
mdape = np.median((np.abs(np.subtract(y_test, y_pred)/ y_test))) * 100
R2 = r2_score(y_test, y_pred)

#create dataframe for the statistical scores
scores = {'score
name':['mse','rmse','stderr','mean_average_error','mape','mdape',"R2"],'Value'
:[mse,rmse,stderr,mean_average_error,mape,mdape,R2]}
df_scores = pd.DataFrame(scores)
print(df_scores)

# save documents to CSV format
test_filepath = 'Z:/Private/f.berks/Thesis/Data/tests/'
results_filepath = str(test_filepath + model_name + "V" + version +
'_results'+'.csv')
results.to_csv(results_filepath, sep=',')
scores_filepath = str(test_filepath + model_name + "V" + version +
'_scores'+'.csv')
df_scores.to_csv(scores_filepath, sep=',')
feature_importance_filepath = str(test_filepath + model_name + "V" + version +
"_featureImportance" +'.csv')
feature_importance.to_csv(feature_importance_filepath, sep=',')
```

## C. Script of Decision Tree algorithm

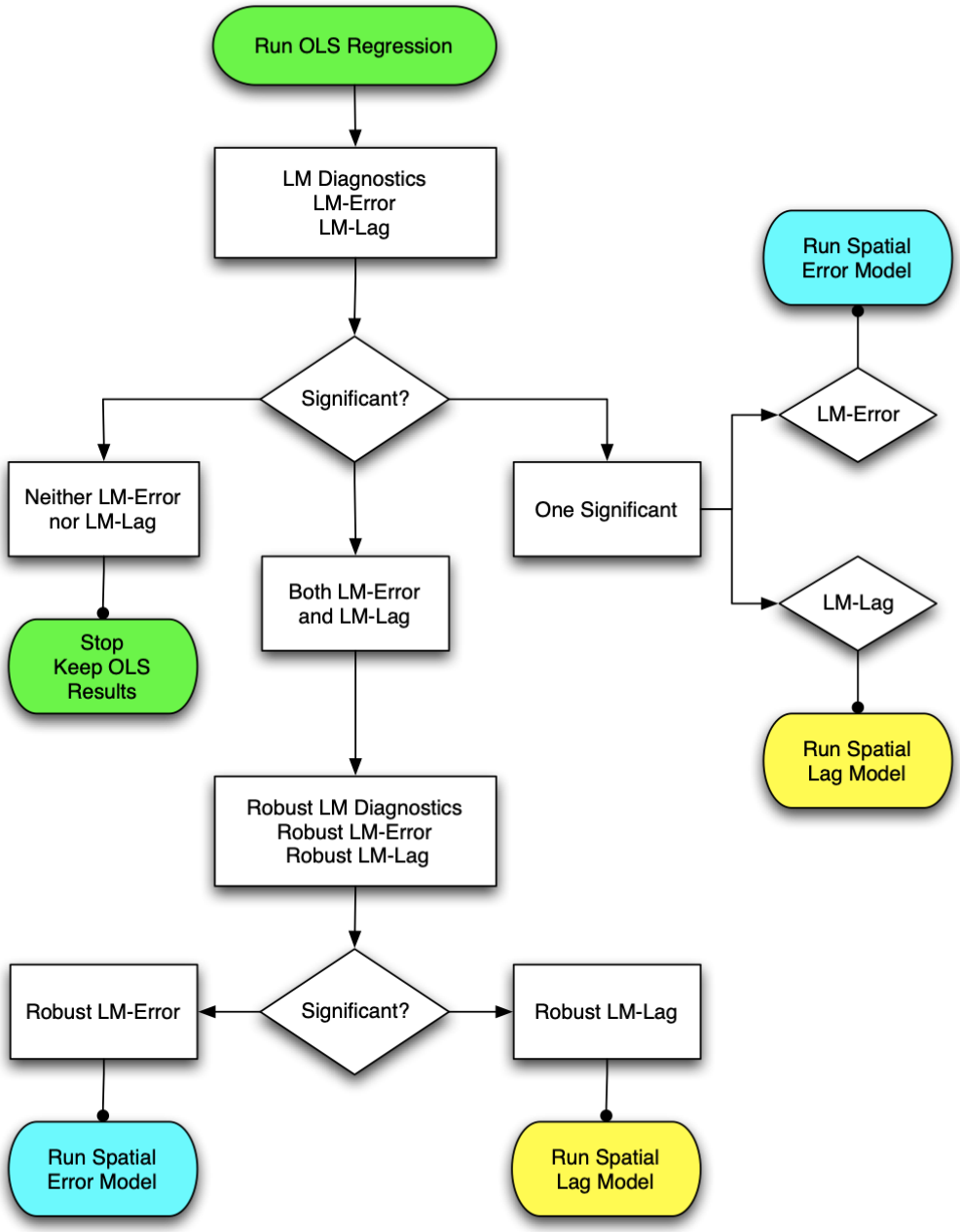
```
import numpy as np
import pandas as pd
import shap
import math
import sklearn

from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_percentage_error
from sklearn.metrics import r2_score

# load the prepared datafile
filepath = 'Z:/Private/f.berks/Thesis/Data/Dataset_CleanedV3.csv'
dataset = pd.read_csv(filepath)

# State name for saving
version = "1.0"
model_name = "DTR"
file_name = model_name+"_"+version
print(file_name)

# dataset sample for faster tweaking and running
# dataset = dataset.sample(100)

# X (Independent variables) and y (target variable)
X = dataset.drop(columns={'Transactionprice_m2'}, axis=1)
y = dataset.Transactionprice_m2

# create headers for the Shap values outcomes
headers_list = list(X.columns)
headers_list_shap = ['shap_'+ item for item in headers_list]

#Splitting the data into train,test data
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=
0)

# define the model
model = DecisionTreeRegressor(random_state=0)
# train the model
model.fit(X_train,y_train)

# run the model
y_pred = model.predict(X_test)

# Create Tree Explainer object that can calculate shap values
```

```
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X_test,approximate=True)
#get the feature importance values from the model
importances = model.feature_importances_

#create a dataframe with the feature names and importance values
feature_importance = pd.DataFrame({'feature': X.columns, 'DTRimportance':
importances})

#concatenate the two tables with the predicted and actual values.
X_test['actual value'] = y_test
X_test['predicted value'] = y_pred
X_test['prediction error posneg'] = (y_test-y_pred)/y_test
X_test['prediction error abs'] = abs(y_test-y_pred)/y_test
X_test[headers_list_shap] = shap_values
results = X_test

# run statistical scores to assess the performance of the model
mse = mean_squared_error(y_test, y_pred)
rmse = math.sqrt(mean_squared_error(y_test, y_pred))
stderr = rmse / math.sqrt(len(y))
mean_average_error = sum(abs(y_pred - y_test)) / len(y_test)
mape = mean_absolute_percentage_error(y_test, y_pred)
mdape = np.median((np.abs(np.subtract(y_test, y_pred)/ y_test))) * 100
R2 = r2_score(y_test, y_pred)

#create dataframe for the statistical scores
scores = {'score
name':['mse','rmse','stderr','mean_average_error','mape','mdape',"R2"],'Value'
:[mse,rmse,stderr,mean_average_error,mape,mdape,R2]}
df_scores = pd.DataFrame(scores)
print(df_scores)

# save documents to CSV format
test_filepath = 'Z:/Private/f.berks/Thesis/Data/tests/'
results_filepath = str(test_filepath + model_name + "V" + version +
'_results'+'.csv')
results.to_csv(results_filepath, sep=',')
scores_filepath = str(test_filepath + model_name + "V" + version +
'_scores'+'.csv')
df_scores.to_csv(scores_filepath, sep=',')
feature_importance_filepath = str(test_filepath + model_name + "V" + version +
"_featureImportance" +'.csv')
feature_importance.to_csv(feature_importance_filepath, sep=',')
```

# D. Arcgis Model builder for hot-cold spot maps

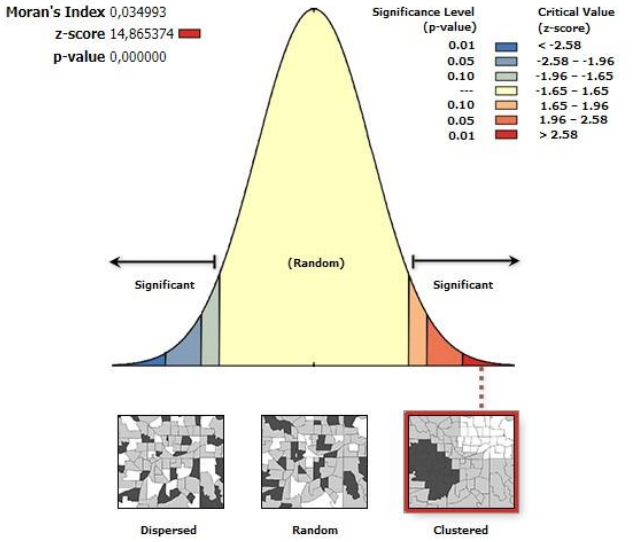


## E. Spatial regression workflow (Anselin, 2005)



## F. Moran's I report Gradient Boosting algorithm

### Spatial Autocorrelation Report




Given the z-score of 14.865374, there is a less than 1% likelihood that this clustered pattern could be the result of random chance.

### Global Moran's I Summary

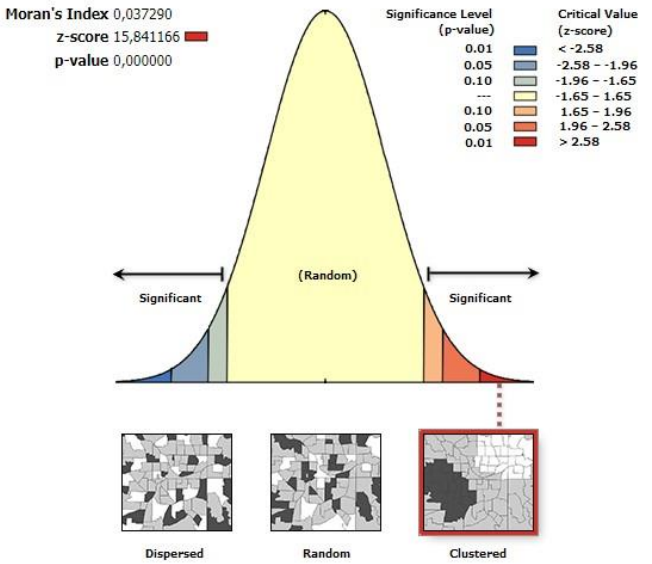
| | |
|---|---|
| **Moran's Index** | 0,034993 |
| **Expected Index** | -0,000224 |
| **Variance** | 0,000006 |
| **z-score** | 14,865374 |
| **p-value** | 0,000000 |

### Dataset Information

| | |
|---|---|
| **Input Feature Class:** | GBmn |
| **Input Field:** | PREDICTION_ERROR |
| **Conceptualization:** | INVERSE_DISTANCE |
| **Distance Method:** | EUCLIDEAN |
| **Row Standardization:** | True |
| **Distance Threshold:** | 20565,4657 Meters |
| **Weights Matrix File:** | None |
| **Selection Set:** | False |

## G. Moran's I report Random Forest algorithm

### Spatial Autocorrelation Report




Given the z-score of 15.841166, there is a less than 1% likelihood that this clustered pattern could be the result of random chance.

### Global Moran's I Summary

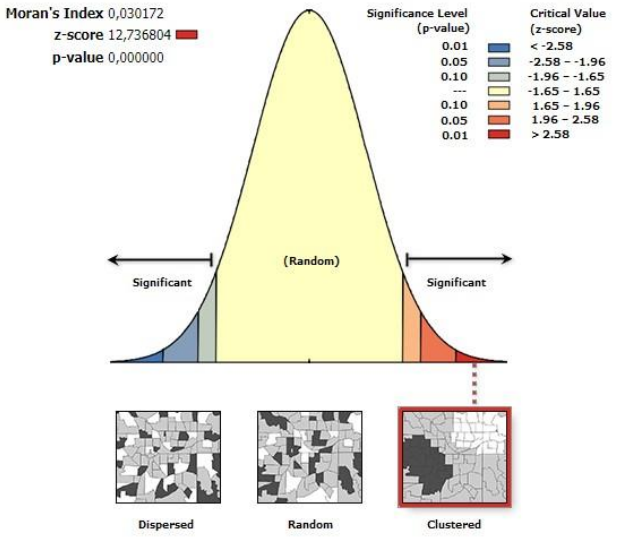
| | |
|---|---|
| **Moran's Index** | 0,037290 |
| **Expected Index** | -0,000224 |
| **Variance** | 0,000006 |
| **z-score** | 15,841166 |
| **p-value** | 0,000000 |

### Dataset Information

| | |
|---|---|
| **Input Feature Class:** | RFmn |
| **Input Field:** | PREDICTION_ERROR |
| **Conceptualization:** | INVERSE_DISTANCE |
| **Distance Method:** | EUCLIDEAN |
| **Row Standardization:** | True |
| **Distance Threshold:** | 20565,4657 Meters |
| **Weights Matrix File:** | None |
| **Selection Set:** | False |

## H. Moran's I report Decision Tree algorithm

### Spatial Autocorrelation Report




Given the z-score of 12.736804, there is a less than 1% likelihood that this clustered pattern could be the result of random chance.

### Global Moran's I Summary

| | |
|---|---|
| **Moran's Index** | 0,030172 |
| **Expected Index** | -0,000224 |
| **Variance** | 0,000006 |
| **z-score** | 12,736804 |
| **p-value** | 0,000000 |

### Dataset Information

| | |
|---|---|
| **Input Feature Class:** | DTRmn |
| **Input Field:** | PREDICTION_ERROR |
| **Conceptualization:** | INVERSE_DISTANCE |
| **Distance Method:** | EUCLIDEAN |
| **Row Standardization:** | True |
| **Distance Threshold:** | 20565,4657 Meters |
| **Weights Matrix File:** | None |
| **Selection Set:** | False |

# I. OLS + spatial error model Random Forest algorithm

```
SUMMARY OF OUTPUT: ORDINARY LEAST SQUARES ESTIMATION
Data set            :  RFmn
Dependent Variable  :  prediction_error
Number of Observations:15232
Mean dependent var  :   0.0336489  Number of Variables   :   10
S.D. dependent var  :     0.08237  Degrees of Freedom    :15222

R-squared           :    0.432406  F-statistic           :      1288.5
Adjusted R-squared  :    0.432071  Prob(F-statistic)     :           0
Sum squared residual:     58.6587  Log likelihood        :     20727.3
Sigma-square        :  0.00385355  Akaike info criterion :    -41434.5
S.E. of regression  :    0.062077  Schwarz criterion     :    -41358.2
Sigma-square ML     :  0.00385102
S.E of regression ML:   0.0620566

------------------------------------------------------------------------------
      Variable      Coefficient      Std.Error    t-Statistic   Probability
------------------------------------------------------------------------------
          CONSTANT   0.000776437    0.000597218        1.30009     0.19356
        Join_Count  -0.000934845    0.000135249       -6.91206     0.00000
              RD_X  -6.56883e-08    1.93474e-08       -3.39521     0.00069
              RD_Y   3.18923e-07    1.22656e-08        26.0014     0.00000
            Spread__  -5.29411e-06    0.000133273     -0.0397239     0.96602
        Floor_Area   0.000301865    1.96559e-05        15.3575     0.00000
    Quality_inside   -0.00873129    0.000748288       -11.6684     0.00000
        PopDensity  -2.68112e-06    4.04144e-07       -6.63408     0.00000
               WOZ    2.6756e-05    8.31269e-06         3.2187     0.00129
       x0_Detached    0.00719594     0.00241742         2.9767     0.00292
------------------------------------------------------------------------------

REGRESSION DIAGNOSTICS
MULTICOLLINEARITY CONDITION NUMBER   19.504199
TEST ON NORMALITY OF ERRORS
TEST                   DF          VALUE            PROB
Jarque-Bera             2       85468735.8739          0.00000

DIAGNOSTICS FOR HETEROSKEDASTICITY
RANDOM COEFFICIENTS
TEST                   DF          VALUE            PROB
Breusch-Pagan test      9       118513.0774          0.00000
Koenker-Bassett test    9          644.1069          0.00000

DIAGNOSTICS FOR SPATIAL DEPENDENCE
FOR WEIGHT MATRIX : RFmn
   (row-standardized weights)
TEST                          MI/DF        VALUE          PROB
Moran's I (error)             0.0222        4.6697        0.00000
Lagrange Multiplier (lag)        1          0.0249        0.87451
Robust LM (lag)                  1         42.3900        0.00000
Lagrange Multiplier (error)      1         21.4311        0.00000
Robust LM (error)                1         63.7961        0.00000
Lagrange Multiplier (SARMA)      2         63.8211        0.00000
============================== END OF REPORT ==============================
```

```
SUMMARY OF OUTPUT: SPATIAL ERROR MODEL - MAXIMUM LIKELIHOOD ESTIMATION
Data set            : RFmn
Spatial Weight      : RFmn
Dependent Variable  :  prediction_error
Number of Observations:15232
Mean dependent var  :    0.033649  Number of Variables   :   10
S.D. dependent var  :    0.082370  Degrees of Freedom    :15222
Lag coeff. (Lambda) :    0.062996

R-squared           :    0.433600  R-squared (BUSE)      : -
Sq. Correlation     : -            Log likelihood        :20737.955353
Sigma-square        :  0.00384292  Akaike info criterion :    -41455.9
S.E of regression   :   0.0619913  Schwarz criterion     :    -41379.6

------------------------------------------------------------------------------
      Variable      Coefficient      Std.Error      z-value     Probability
------------------------------------------------------------------------------
          CONSTANT   0.000684119    0.000628008        1.08935     0.27600
        Join_Count  -0.000940913    0.000136208       -6.9079      0.00000
              RD_X  -7.05467e-08    1.98612e-08       -3.55199     0.00038
              RD_Y   3.2677e-07    1.24895e-08        26.1635     0.00000
            Spread__  -1.18364e-05    0.000133325     -0.088778    0.92926
        Floor_Area   0.000301651    1.96381e-05        15.3605     0.00000
    Quality_inside   -0.0090456    0.000752926       -12.0139     0.00000
        PopDensity  -2.74268e-06    4.09029e-07       -6.70533     0.00000
               WOZ   2.70645e-05    8.42372e-06        3.21289     0.00131
       x0_Detached    0.00669168       0.002418        2.76744     0.00565
            LAMBDA     0.0629965       0.013381         4.7079     0.00000
------------------------------------------------------------------------------

REGRESSION DIAGNOSTICS
DIAGNOSTICS FOR HETEROSKEDASTICITY
RANDOM COEFFICIENTS
TEST                                   DF      VALUE          PROB
Breusch-Pagan test                      9    118781.0391      0.00000

DIAGNOSTICS FOR SPATIAL DEPENDENCE
SPATIAL ERROR DEPENDENCE FOR WEIGHT MATRIX : RFmn
TEST                                   DF      VALUE          PROB
Likelihood Ratio Test                   1        21.4044      0.00000
============================== END OF REPORT ==============================
```

## J. OLS + spatial error model Decision Tree algorithm

```
SUMMARY OF OUTPUT: ORDINARY LEAST SQUARES ESTIMATION
Data set            :  DTRmn
Dependent Variable  :  prediction_error
Number of Observations:15232
Mean dependent var  :    0.0494479  Number of Variables   :   10
S.D. dependent var  :     0.112554  Degrees of Freedom    :15222

R-squared           :     0.491818  F-statistic           :     1636.87
Adjusted R-squared  :     0.491517  Prob(F-statistic)     :           0
Sum squared residual:       98.061  Log likelihood        :     16813.7
Sigma-square        :  0.00644206  Akaike info criterion :    -33607.5
S.E. of regression  :    0.0802624  Schwarz criterion     :    -33531.2
Sigma-square ML     :  0.00643783
S.E of regression ML:    0.0802361

------------------------------------------------------------------------------
       Variable      Coefficient      Std.Error    t-Statistic    Probability
------------------------------------------------------------------------------
          CONSTANT    0.000925937    0.000772174        1.19913      0.23045
        Join_Count    -0.00176076     0.00017487       -10.069      0.00000
              RD_X  -7.04471e-08    2.50152e-08       -2.81617      0.00487
              RD_Y   4.01639e-07    1.58588e-08        25.3259      0.00000
           Spread__  -0.000643546    0.000172315       -3.73471      0.00019
        Floor_Area    0.000297988    2.54141e-05        11.7253      0.00000
    Quality_inside    -0.0054492    0.000967499       -5.63225      0.00000
        PopDensity  -3.69464e-06    5.22538e-07       -7.07056      0.00000
               WOZ   4.62873e-05    1.07479e-05        4.30664      0.00002
       x0_Detached     0.0100632     0.00312561         3.2196      0.00129
------------------------------------------------------------------------------

REGRESSION DIAGNOSTICS
MULTICOLLINEARITY CONDITION NUMBER   19.504199
TEST ON NORMALITY OF ERRORS
TEST                     DF           VALUE              PROB
Jarque-Bera               2       13267952.1065          0.00000

DIAGNOSTICS FOR HETEROSKEDASTICITY
RANDOM COEFFICIENTS
TEST                     DF           VALUE              PROB
Breusch-Pagan test        9         75783.5427          0.00000
Koenker-Bassett test      9          1040.4251          0.00000

DIAGNOSTICS FOR SPATIAL DEPENDENCE
FOR WEIGHT MATRIX : DTRmn
   (row-standardized weights)
TEST                           MI/DF        VALUE          PROB
Moran's I (error)              0.0201        4.2408        0.00002
Lagrange Multiplier (lag)         1          1.0442        0.30684
Robust LM (lag)                   1         51.0507        0.00000
Lagrange Multiplier (error)       1         17.6456        0.00003
Robust LM (error)                 1         67.6521        0.00000
Lagrange Multiplier (SARMA)       2         68.6963        0.00000
============================== END OF REPORT ==============================
```

```
SUMMARY OF OUTPUT: SPATIAL ERROR MODEL - MAXIMUM LIKELIHOOD ESTIMATION
Data set            : DTRmn
Spatial Weight      : DTRmn
Dependent Variable  :  prediction_error
Number of Observations:15232
Mean dependent var  :    0.049448  Number of Variables   :   10
S.D. dependent var  :    0.112554  Degrees of Freedom    :15222
Lag coeff. (Lambda) :    0.057021

R-squared           :    0.492694  R-squared (BUSE)      : -
Sq. Correlation     : -            Log likelihood        :16822.529449
Sigma-square        :  0.00642672  Akaike info criterion :    -33625.1
S.E of regression   :   0.0801668  Schwarz criterion     :    -33548.7

------------------------------------------------------------------------------
       Variable      Coefficient      Std.Error      z-value    Probability
------------------------------------------------------------------------------
          CONSTANT    0.000789926    0.000808028       0.977597      0.32827
        Join_Count    -0.00175299    0.000176006       -9.95979      0.00000
              RD_X  -7.71022e-08    2.56177e-08       -3.00972      0.00261
              RD_Y   4.11196e-07    1.61221e-08        25.5052      0.00000
           Spread__  -0.000656421    0.000172392       -3.80771      0.00014
        Floor_Area    0.000298167    2.53956e-05        11.7409      0.00000
    Quality_inside    -0.00577614    0.000973019       -5.93631      0.00000
        PopDensity  -3.77743e-06    5.28293e-07       -7.15027      0.00000
               WOZ   4.56431e-05    1.08787e-05        4.19565      0.00003
       x0_Detached     0.00954099     0.00312657        3.05158      0.00228
            LAMBDA      0.0570213      0.0134119        4.25153      0.00002
------------------------------------------------------------------------------

REGRESSION DIAGNOSTICS
DIAGNOSTICS FOR HETEROSKEDASTICITY
RANDOM COEFFICIENTS
TEST                                   DF       VALUE          PROB
Breusch-Pagan test                     9      75958.6672      0.00000

DIAGNOSTICS FOR SPATIAL DEPENDENCE
SPATIAL ERROR DEPENDENCE FOR WEIGHT MATRIX : DTRmn
TEST                                   DF       VALUE          PROB
Likelihood Ratio Test                  1         17.5705      0.00003
============================== END OF REPORT ==============================
```

## K. OLS + spatial error model Gradient Boosting algorithm

```
SUMMARY OF OUTPUT: ORDINARY LEAST SQUARES ESTIMATION
Data set            :  GBmn
Dependent Variable  :  prediction_error
Number of Observations:15232
Mean dependent var  :   0.0379884  Number of Variables   :   10
S.D. dependent var  :   0.0910561  Degrees of Freedom    :15222

R-squared           :    0.444764  F-statistic           :     1354.82
Adjusted R-squared  :    0.444436  Prob(F-statistic)     :           0
Sum squared residual:     70.1217  Log likelihood        :     19367.8
Sigma-square        :   0.0046066  Akaike info criterion :    -38715.6
S.E. of regression  :    0.067872  Schwarz criterion     :    -38639.3
Sigma-square ML     :  0.00460358
S.E of regression ML:   0.0678497

------------------------------------------------------------------------------
       Variable      Coefficient      Std.Error    t-Statistic   Probability
------------------------------------------------------------------------------
          CONSTANT   0.000767506     0.00065297        1.17541     0.23979
        Join_Count  -0.000689636    0.000147874       -4.66367     0.00000
              RD_X  -1.17431e-08    2.11535e-08      -0.555139     0.57862
              RD_Y   2.77458e-07    1.34106e-08        20.6894     0.00000
            Spread__   0.000226719   0.000145714        1.55592     0.11973
        Floor_Area   0.000214081    2.14908e-05        9.96152     0.00000
    Quality_inside   -0.00631833    0.000818142       -7.72278     0.00000
        PopDensity  -2.63288e-06    4.41872e-07       -5.95846     0.00000
               WOZ   6.55726e-05    9.08869e-06        7.21474     0.00000
      x0_Detached     0.0242844     0.00264309        9.18787     0.00000
------------------------------------------------------------------------------

REGRESSION DIAGNOSTICS
MULTICOLLINEARITY CONDITION NUMBER   19.504199
TEST ON NORMALITY OF ERRORS
TEST                   DF           VALUE             PROB
Jarque-Bera             2        87334667.5642          0.00000

DIAGNOSTICS FOR HETEROSKEDASTICITY
RANDOM COEFFICIENTS
TEST                   DF           VALUE             PROB
Breusch-Pagan test      9         68364.6591          0.00000
Koenker-Bassett test    9           367.5474          0.00000

DIAGNOSTICS FOR SPATIAL DEPENDENCE
FOR WEIGHT MATRIX : GBmn
   (row-standardized weights)
TEST                          MI/DF        VALUE          PROB
Moran's I (error)             0.0203        4.2751        0.00002
Lagrange Multiplier (lag)       1           0.1332        0.71513
Robust LM (lag)                 1          29.5745        0.00000
Lagrange Multiplier (error)     1          17.9346        0.00002
Robust LM (error)               1          47.3759        0.00000
Lagrange Multiplier (SARMA)     2          47.5091        0.00000
============================== END OF REPORT ==============================
```

```
SUMMARY OF OUTPUT: SPATIAL ERROR MODEL - MAXIMUM LIKELIHOOD ESTIMATION
Data set            : GBmn
Spatial Weight      : GBmn
Dependent Variable  :  prediction_error
Number of Observations:15232
Mean dependent var  :    0.037988  Number of Variables   :   10
S.D. dependent var  :    0.091056  Degrees of Freedom    :15222
Lag coeff. (Lambda) :    0.057863

R-squared           :    0.445747  R-squared (BUSE)      : -
Sq. Correlation     : -            Log likelihood        :19376.811787
Sigma-square        :  0.00459543  Akaike info criterion :    -38733.6
S.E of regression   :   0.0677896  Schwarz criterion     :    -38657.3

------------------------------------------------------------------------------
       Variable      Coefficient      Std.Error       z-value    Probability
------------------------------------------------------------------------------
          CONSTANT   0.000679968    0.000683761       0.994452     0.32000
        Join_Count  -0.000697312    0.000148849       -4.68471     0.00000
              RD_X  -1.52681e-08    2.16704e-08      -0.704559     0.48108
              RD_Y   2.83784e-07    1.36364e-08        20.8107     0.00000
            Spread__   0.000216748   0.000145779        1.48683     0.13706
        Floor_Area   0.000212154    2.14747e-05        9.87923     0.00000
    Quality_inside   -0.00657115    0.000822871       -7.98564     0.00000
        PopDensity  -2.69782e-06    4.46807e-07       -6.03799     0.00000
               WOZ   6.67772e-05    9.20086e-06        7.25772     0.00000
      x0_Detached     0.0238964     0.00264389        9.03835     0.00000
            LAMBDA     0.057863      0.0134076        4.31568     0.00002
------------------------------------------------------------------------------

REGRESSION DIAGNOSTICS
DIAGNOSTICS FOR HETEROSKEDASTICITY
RANDOM COEFFICIENTS
TEST                                   DF       VALUE          PROB
Breusch-Pagan test                     9      68510.8947      0.00000

DIAGNOSTICS FOR SPATIAL DEPENDENCE
SPATIAL ERROR DEPENDENCE FOR WEIGHT MATRIX : GBmn
TEST                                   DF       VALUE          PROB
Likelihood Ratio Test                  1         17.9767      0.00002
============================== END OF REPORT ==============================
```