university of groningen / faculty of spatial sciences

Master Thesis
Economic Geography

# Geography and AI: a happy marriage? Exploring the potential of Machine Learning as a new method in geographic research.


*Author*
I.P.S. Kema

*Supervisors*
dr. S. Koster
prof. dr. D. Ballas


August 28, 2020

## ABSTRACT

Big data analytics can offer Geography new ways of understanding complex socio-spatial processes, especially with the increasingly available amount of data that is produced by society. This thesis explores the potential of machine learning in Geography via a case study of neighbourhood level gentrification prediction. Several machine learning algorithms are compared; XGBoost, CatBoost, and Random Forest regression outperform standard quantitative methods. The implementation of SHapley Additive exPlanations (SHAP) as a way of interpreting machine learning models is explored, and suggests that SHAP is a promising solution to the need for explainable machine learning models. Future gentrification prediction reveals that model specification has substantial impact on result interpretation and practical applicability, and suggests that theoretical foundation remains a key factor in future development of the research field. Machine learning provides a lot of new opportunities for geography but it is also important to be critical of its promises.

# PREFACE

I would first like to thank my thesis supervisor dr. Sierdjan Koster for his guidance, enthusiasm and commitment; even during a global pandemic. Your advice and feedback were invaluable.

I would also like to acknowledge prof. dr. Dimitris Ballas as the second reader of this thesis, and I am grateful for his help.

Finally, I would like to thank my parents for their love and support.

# CONTENTS

# 1

# INTRODUCTION

The increasingly available amount of data that our society produces presents many new opportunities for quantitative geographic research. Kitchin (2014, p. 5) characterizes this as a Data Revolution and typifies the observation as the emergence of data-driven science, which "seeks to hold to the tenets of the scientific method, but is more open to using a hybrid combination of abductive, inductive and deductive approaches to advance the understanding of a phenomenon". The aim of data-driven science is to generate insights 'born from data' instead of the more conventional 'born from theory' approach.

The field of Artificial Intelligence is essential to Big Data analytics (also referred to as Data Science). Machine Learning algorithms are traditionally used for tasks such as Computer Vision (e.g. object detection), Natural Language Processing, and Recommender Systems. Engagement by "mainstream" data scientists with geographical methods and thinking has been fairly minimal to date (Arribas-Bel and Reades, 2018), although quite a lot of the Big Data generated by our society is spatially embedded and that geographical traditions may have much to offer to "Big Data" research (Arribas-Bel and Reades, 2018). Applying Data Science methods to the field of Geography can offer us new ways of modelling complex socio-spatial processes, which can result in novel and potentially fundamentally different insights into both new and already established work within the geographic domain. Big Data analytics enables an entirely new epistemological approach for making sense of the world; rather than testing a theory by analyzing relevant data, new data analytics seek to gain insights 'born from the data' (Kitchin, 2014). Singleton and Arribas-Bel (2019) argue that there is substantial potential for the establishment of a Geographic Data Science within Geography, noting benefits for the scientific field in terms of being able to implement more effective, ethical, and epistemologically robust analytics; as well as "sustaining the relevance of Geography and subdisciplinary approaches within a rapidly changing socio-technological landscape".

An important question that arises is how complex it is to obtain the aforementioned "more effective, ethical, and epistemologically robust analytics". When do we deem a geographic analysis as robust? In what way is this robustness limited in terms of data, and how does this differ between methods? Due to how new Geographic Data Science is, there is barely any domain knowledge present on how machine learning methods perform when it comes to analyzing and predicting social phenomena. The work of Reades et al. (2019) aims to understand urban gentrification in London with the use of machine learning. Their analysis shows that the chosen machine learning algorithm outperforms traditional quantitative methods. Since their data and code are publicly available this makes their research a worthwhile candidate for analyzing the robustness of machine algorithms and data in geographic research.

Social relevance of this research can be found in that it attempts to improve understanding of gentrification in a methodical way. Although the topic of gentrification is chosen to function as an illustration of applied machine learning, research on gentrification is important for society because

deeper knowledge of how and why urban areas experience economic uplift or decline can play an important role in society's economic and social policy formation. Gentrification is on one hand characterized by economic improvement in a certain neighborhood, suggesting the phenomenon as a positive one, yet this is also associated with displacement of original inhabitants and an increase of social inequality. If the methods outlined in this thesis outperform current methods with less data, this would allow geographers to identify gentrification easier and improve the decision-making process for policy makers. Academic relevance of this thesis is twofold. It is interesting in a technological sense: how does Machine learning perform in the geographic domain algorithmically, and which model approach produces optimal results? This provides AI researchers with new understanding and insight on the capabilities and limitations of machine learning in non-traditional research domains, and can potentially contribute to the development of new machine learning methods. On the other hand, this type of research is also interesting from a geographical perspective. What insights can we generate through machine learning that we cannot obtain through more conventional research methods within the field of Geography? These new insights can help geographers with gaining new, different, and potentially better understanding of socio-spatial phenomena. For example, machine learning can contribute to unlocking big data analytics for geographic research, which makes it possible to perform research in a new way. The topic of gentrification lends itself well to this aim, because it is a complex field of study that would benefit from new quantitative approaches. Easton et al. (2019) for example remark that due to the multidimensional character of gentrification, it would be preferable to identify neighbourhoods undergoing gentrification using sensitivity testing for different univariate proxies.

Due to how new the field of research is— (Arribas-Bel and Reades, 2018) propose it to be called Geographic Data Science (GDS) but it's also referred to in literature as Geocomputation—, we do not have any state-of-the-art approaches or best practices available yet, apart from Reades et al. (2019) own experimentation and a select few other exploratory approaches. This means that on one hand there is not much existing research to go by, but it also means that there's a lot of low hanging fruit; a lot of new potentially valuable insights can be acquired relatively easily. Experimenting with feature selection of variables in combination with machine learning can also potentially benefit the field of Geography a lot. Another more practical potential benefit could be that in future geographic research, better results can be obtained with less (or less complete) data sets, which makes it easier to perform quantitative humanities research.

The goal of this Master's thesis is to perform a robustness analysis on predicting and understanding gentrification at a neighborhood level with the use of machine learning, and to expand upon this approach in a technical way and on a conceptual level. Since geography is a non-traditional application field for AI models and is primarily concerned with relatively complex social phenomena, interpretation of outcome therefore requires more detailed understanding. Finding what is being modelled exactly in geographic machine learning, and what it means for the usefulness of such an approach are important questions to explore when moving towards an integrated field of Geographic Data Science as defined by Arribas-Bel and Reades (2018). This analysis on gentrification will provide results that al-

low us to obtain new insight into the utility of machine learning as a new methodological approach in geographic research.

The availability of data and code as outlined in Reades et al. (2019) makes it possible to reproduce work and allows us to create new domain knowledge by changing data, parameters, and predictive algorithm. Generating furthering technical understanding is to be done through experimentation with several alternative machine learning algorithms such as Support Vector Machine (SVM), K-Nearest Neighbors, or Boosting algorithms. Finding out the importance of used gentrification variable data will be achieved through a feature analysis with SHapley Addition eXplanations (SHAP) (Lundberg and Lee, 2017). The main research gap that can be identified for this thesis is that although AI appears to generate promising results for geographic topics such as gentrification, there has not yet been a critical evaluation of these results and used methods.

This brings us to the following research question: how robust is Machine Learning in the prediction and understanding of gentrification? The application of machine learning algorithms and data science methods in the field of Geography are rather sparse in the current published literature. As such, there are questions pertaining to its effectiveness in explaining social phenomena, the added value of these novel techniques compared to traditional quantitative approaches, and its implications for the development of economic and spatial policy. The already performed research by Reades et al. (2019) on gentrification of London neighborhoods provides us with an excellent starting point for the answering of these questions, and allows us to further explore the algorithm side as well as the potential practical value of machine learning in geographic research. For this we define the following four sub-questions:

1. In what way does our data impact machine learning decision-making compared to more conventional regression approaches? A feature importance analysis with SHAP allows us to find out which variables are most important to include when it comes to predicting gentrification at neighborhood level, and whether this combination of variables/features is consistent across different model implementations.

2. Which algorithm is best suited for prediction of gentrification? (Random Forest, KNN, SVM, Regression, Boosting algorithms?) Currently in the literature only a tuned version of a Random Forest algorithm is compared to linear regressions for gentrification prediction. It will be interesting to see if better prediction scores can be achieved with different machine learning algorithms, such as Support Vector Machines (SVM), K-Nearest Neighbors, and Tree Boosting. Linear regression will be used as a benchmark. In addition, the algorithm comparison enables us to compare model sensitivity of feature importance and of future prediction.

3. Do comparable performance results also translate to comparable predictions of future neighborhood change? If we visualize gentrification results from the machine learning model, are we able to observe any distinct gentrification patterns?

4. What potential new domain knowledge can we obtain from this analysis? What new findings or insights to be gained in terms of predictive method and data considerations, and how do they help us in making machine learning applied to social science work?

The thesis is set up in the following way: 1) theoretical framework, 2) method explanation, 3) results & discussion, 4) concluding remarks. The theoretical chapter contains a section on the definition and measuring of gentrification, an overview of machine learning in general, explanation of used algorithms, and a section on relevant applied machine learning research literature. The method chapter outlines data used, machine learning analysis, feature importance explanation and evaluation metrics. The results & discussion section contain machine learning performance results, SHAP results, and gentrification map prediction and comparison.

# 2

# THEORETICAL BACKGROUND

This thesis is positioned at an intersection of two different fields of study: Human Geography and Data Science. Consequently, this theoretical chapter will focus on explaining several concepts, theories and definitions of both fields so readership from both fields will be able to sufficiently grasp both the technical underpinnings and domain knowledge background. Additionally, we will examine the implications of Machine Learning on the production of geographic knowledge.

The first section contains the definitions of gentrification and how it is quantified within scientific literature. This is because if we want to let a machine learning algorithm predict gentrification, we have to understand the conceptual basis of first in order to adequately valuate such an approach. Social concepts such as gentrification generally have a high degree of complexity to them, and often there is no one agreed upon definition within the research field. Having sufficient insight into the concept is therefore crucial when it comes to interpreting and evaluating machine learning model results, or else we risk the spatial aspect "to be rationalized only as a supplementary column within a database, no more or less important than any other attribute" (Singleton and Arribas-Bel, 2019). Domain knowledge is incredibly important when it comes to selecting variable data in machine learning (Guyon and Elisseeff, 2003), so finding out from literature which variables are particularly important when it comes to neighbourhood gentrification will also be done in this section. Secondly, this theoretical chapter will contain a primer on machine learning and related data science concepts. This will include a basic overview of how machine learning works and a conceptual explanation of the different machine learning algorithms used in this research. We will also go over different machine learning implementations in social science literature, in order to compile machine learning domain knowledge potentially relevant for the field of geography. The last section details the considerations and implications of Machine Learning applied to Geography. What is the added value of Data Science/ML as a tool for geographic research, and which ontological challenges need to be resolved in order to make this approach work?

## 2.1 GENTRIFICATION

The term gentrification was first coined by Ruth Glass in the early 1960s as she observed the arrival of the 'gentry' and the accompanying social transition of several districts in central London (Gregory et al., 2011). The term was used to describe the London middle and upper classes moving into the traditionally working-class neighbourhoods, with as a result the displacement of incumbent residents and change of social character of the neighbourhood. From this definition two important components can be defined: 1) gentrification raises the economic level of a neighborhood population, 2) gentrification changes a neighborhood's social character or culture. These components are important because they helped shape later definitions (Barton, 2016). Over the years multiple definitions of gentrification have been formulated, varying in conceptual focus and complexity, as well

as opposing views of its effect on society. Gentrification is contested and controversial. There are political and academic opponents—as well as advocates—of the process (Helbrecht, 2018). Rigolon and Németh (2019) find that explanations for gentrification vary widely from political–economic/supply-side/production- oriented perspectives (Harvey, 1985; Smith and Sorkin, 1992; Smith, 2005) to social–cultural/demand-side/consumption-oriented perspectives (Caulfield, 1994; Ley, 1994; Rose, 1996; Helbrecht, 2018). Positive impacts of gentrification include new investment in areas, service improvement, and creation of new jobs. A primary negative effect is that 'original' neighborhood inhabitants are forced out of the gentrifying neighbourhood, either directly through policy implementation or indirectly as a result of an increased cost of living. House value goes up due to newly increased demand so rent prices adjust accordingly. Thus, gentrification is quite often seen as a displacement process that segregates the social strata of a city along the social-spatial axis of wealth (Helbrecht, 2018). The research community now generally accepts that these competing explanations are better understood as representing ends of a continuum and that both production and consumption perspectives are crucially important in explaining, understanding, and dealing with gentrification (Rigolon and Németh, 2019).

### 2.1.1 *Measuring Gentrification*

Holm and Schulz (2018) observe that after more than 50 years of gentrification research there is still no consensus about a measurement tool for gentrification. This they primarily attribute to the lack of agreement on a definition of gentrification. Definitions that have been used to identify gentrification areas in empirical studies usually vary based on the selected methodology. According to Barton (2016), in qualitative studies researched neighbourhoods are frequently selected based on cultural changes due to demographic shifts in neighbourhood populations, with much of the research placing an emphasis on a transition from racial or ethnic neighbourhood cultures to middle-class, white culture (Anderson, 2013; Maurrasse, 2014). This shift in culture and demography was often related with change in housing and local business. In quantitative studies on the other hand, the change in the neighbourhood's socio-demographic structure is considered to be the key criterion for gentrification processes (Barton, 2016). Quantitative studies often use a threshold strategy where neighbourhoods were identified as gentrifiable if they featured a particular characteristic or characteristics at the beginning of a decade and gentrified if the characteristic changed in particular way. Both qualitative and quantitative approaches have their advantages, as well as their shortcomings. For example, qualitative data will tend to be richer in terms of research detail and explanatory power (Barton, 2016). This complexity makes it also a lot more difficult to gather a sufficient amount of data; taking into account factors such as physical, economic, social and cultural neighbourhood changes will require data collection for all these factors. This increases complexity for data collection, model building, as well as interpretation of findings. For quantitative approaches it is the other way around: they easier to execute, but quite often lacks depth when it comes to measuring more specific phenomena. Barton (2016) finds that while the definitions used in quantitative strategies were easier to operationalize, they often did not include references to changes in the 'social character' or local culture.

### 2.1.2 *Quantitative neighborhood gentrification*

Although the broad dimensions of gentrification are often agreed on, the operationalization of these dimensions in terms of measurable variables is far more difficult to define without ambiguity (Easton et al., 2019). For example, Galster and Peacock (1986) find that variable selection has a significant impact on which, and how many, census tract areas were identified as experiencing gentrification. Their operationalization approach was to construct several logistic regression models using census variables for Philadelphia (1970-1980). A comparison study by Barton (2016) on gentrification measurement strategies provides similar conclusions. The results were that each of the strategies identified different neighborhoods undergoing gentrification.

Owens (2012) operationalizes neighbourhood gentrification through the concept of Socio-economic Status (SES) as a metric for neighbourhood ascent. Neighbourhood ascent is defined as "neighbourhoods in which, at the aggregate level, residents' income, housing costs, and educational and occupational attainment increased". A metric for neighborhood SES is calculated by combining 5 census data variables: average household income, average house values, average gross rent, proportion of residents over 25 years old with a BA, and proportion of workers over 16 years old working in a managerial, technical, or professional (high-status) job. Principal Component Analysis (PCA) is used to combine many correlated variables into one indicator by assessing the similarities and differences among the variance of each variable (Owens, 2012). Walks and Maaranen (2008) perform a similar experiment to identify gentrification on neighborhood level. They apply PCA to four variables that are assumed to identify both timing and extent of gentrification; average personal income, proportion of tenants, socioeconomic status based on employment rate, and percentage of artists resident in an area. Reades et al. (2019) utilize a Random Forest machine learning model in order to relate neighborhood ascent in London. For operationalization of ascent they follow the method of Owens (2012). Furthermore they include 166 different explanatory variables, including environmental measures such as the amount of green space available, and average travel time to central London. Holm and Schulz (2018) propose a model called GentriMap for measuring gentrification and displacement in Berlin. They define gentrification as the conjunction of social upgrading and real-estate value increases, which in the model is achieved via the construction of a real-estate index and a social index to quantify the respective model components. The real-estate index consists of four indicators: average rental prices offered, average prices for individually owned apartments, the number of apartments offered for rent, and the number of apartments offered for sale. In addition, they included the number of offers as an indicator variable since it provides an indication of the extent of real-estate value increases. For the social index only one indicator variable is used: the number of transfer payment recipients in accordance with the the German Social Insurance Code. This indicator variable includes recipients of different types of welfare benefits, and is interpreted as the lowest estimate of low-income people in an area. The approach operationalizes a relational definition of gentrification, which means that it measures gentrification processes solely in relation to the rest of the city. For this thesis the selected operationalization of neighborhood gentrification is the one defined by Owens (2012), because this enables us to compare results from Reades et al. (2019) since they use this same definition. This comparison is needed in order to answer the formulated research

questions. Using different gentrification metrics such as the one outlined by Holm and Schulz (2018) is a potentially valuable alternative approach, because this will allow comparison at a method level. Unfortunately this method comparison does not fall into the scope of this thesis, and is instead something for future research.

## 2.2 MACHINE LEARNING

### 2.2.1 *Primer: what is Machine Learning?*

According to Jordan and Mitchell (2015, p. 255), Machine Learning is a discipline that is focused on two interrelated questions: "How can one construct computer systems that automatically improve through experience?" and "What are the fundamental statistical, computational, and information-theoretic laws that govern all learning systems, including computers, humans, and organizations?". While this definition does contain the essence of what machine learning encompasses, it is still a quite technical definition and requires some background knowledge of computer science to make you as a reader fully understand. In more simple terms it is the field of study that gives computers the ability to learn without being explicitly programmed; in machine learning computers learn from data instead of executing a rule-based script that is written by a programmer. This is what is meant by the "automatically improve through experience" part by Jordan and Mitchell (2015). At its core computers require explicit instructions by a human in order to function. A computer computes: in other words, it performs a calculation. These machine instructions are written in something called a programming language. A programming language can be implemented at a hardware level: moving of 1s and 0s (binary) in a computer's memory, but it can also be abstracted into a programming language that makes it easier to read and work with for a human (if this condition satisfied, then execute function). This difference in abstraction makes a programming language a low-level or a high-level language. Choice of programming language depends on whether computational performance or programming flexibility is more important.

Machine learning is in a sense somewhat contradictory in its definition: how is it possible for computers to do something without explicit instructions ("learn" from data) when they need instructions to function? This has everything to do with how machine learning differs from "regular" rule-based systems. A rule-based system functions through facts (data values in a database) and user-crafted rules on what to with the data. These rules are constructed to automate a human decision process: if data point exceeds a certain value or matches with another data point, follow specified procedure. Machine learning does not replicate this human specified decision process, but instead "learns" only from the outcome. For example: why a certain e-mail is considered a spam e-mail is not relevant for a machine learning model, it only needs to know that people mark a certain message as spam. The model does not need human-curated rules in order to perform the task, which means it is very flexible in its application. Rule-based systems on the other hand are limited by the fact that they only function within their defined set of rules: dealing with special cases (not specified by the rule set) is very difficult or impossible. Another problem with rule-based systems is that for certain tasks data and domain knowledge change very quickly and/or frequently: they change faster than it takes to update the rule set.

This could also imply a nearly impossibly long set of rules if the task is complex enough. Being able to surpass these challenges that rule-based systems face is what makes machine learning so powerful. Goodfellow et al. (2016) therefore states that "the difficulties faced by systems relying on hard-coded knowledge suggest that AI systems need the ability to acquire their own knowledge, by extracting patterns from raw data". He defines this capability of learning from data as machine learning.

This learning from data approach makes it possible for computers to perform tasks and tackle problems that involve real world understanding. These "real world" problems are, ironically enough, quite difficult for a computer to solve, while for humans they can be almost trivial. The opposite is true for formal, abstract problems: computers are very efficient at calculating things like complex numbers or determining the shortest path between two points on a map, or exactly reproducing information, which is extremely difficult for a human. Why machine learning problems such as object detection, face recognition or speech recognition are so difficult for a computer and easy for a human is because the tasks require a lot of knowledge about the world. Abstract problems such as finding the winning steps for a game of tic-tac-toe are narrowly defined problems, do not contain ambiguity or subjective knowledge, and therefore require only a limited set of rules (knowledge). The human brain on the other hand is able to make sense of this vast amount of subjective and ambiguous knowledge about everything just fine. Goodfellow et al. (2016) remarks that due to the fact that much of this knowledge is subjective and intuitive, it is therefore difficult to articulate in a formal way. And so if we want computers to behave in an intelligent way (perform human tasks & deal with subjectivity) we will need to be able to capture this informal knowledge. This is what machine learning attempts to accomplish (sometimes quite successfully).

Now that we have outlined what machine learning is on a conceptual level, we will now look at a few machine learning algorithms and how these are implemented. Within the field of Machine Learning we make a distinction between different types of learning algorithms. Géron (2017) classifies them in broad categories based on:

- Whether or not they are trained with human supervision (supervised, unsupervised, semi-supervised, and Reinforcement Learning)
- Whether or not they can learn incrementally on the fly (online versus batch learning)
- Whether they work by simply comparing new data points to known data points, or instead detect patterns in the training data and build a predictive model, much like scientists do (instance-based versus model-based learning)

However, these are not exclusive criteria: there exists overlap between the different categories, and it is possible to build a machine learning system that is a combination of them. Although a thorough explanation of all three categories is useful for a deeper understanding of machine learning, in regard to their relevance for this thesis we will primarily focus on this first category. Goodfellow et al. (2016) outlines the following tasks as most common for machine learning:

- Classification: computer program is asked to specify which defined category some input belongs to

- Regression: computer program is asked to predict a numerical value given some input
- Transcription: system is asked to observe an unstructured representation of data and transcribe this into a discrete structured form. Examples: optical character recognition (OCR), Speech recognition where sound data is transcribed into text data.
- Structured output: computer program is asked to output important relations between different elements. Examples: image captioning, natural language sentence parsing.
- Anomaly detection: computer program analyses a set of events and determines unusual occurrences. Examples: spam detections, fraud detection
- Imputation: program is asked to provide a prediction of values of missing entries

The main difference between supervised and unsupervised learning algorithms has primarily to do with how the data is structured. Goodfellow et al. (2016) defines this difference as "by what kind of experience they (the machine learning algorithms) are allowed to have during the learning process". In order to perform supervised learning, the data has to be labelled (also called a target). This labelling is what makes the type of learning "supervised". Unsupervised learning on the other hand does not require labels, but because of this also produces less powerful results. Semi-supervised learning trains on partly labeled data and partly unlabeled data. Labels for the whole dataset are generated using the labeled part. The term supervised learning comes from the view that the label/target is being provided by an instructor (human labelling); this label "supervises" the algorithm's learning. In unsupervised learning the algorithm attempts to make sense of the data without this label guide. Reinforcement learning functions quite differently from the aforementioned types of learning (Géron, 2017). An agent (the reinforcement learning system) learns by performing actions which results either in a reward or a penalty. Over time it performs actions until the best strategy is formed: this strategy is called a policy.

### 2.2.2 *Supervised Learning*

According to Géron (2017) some of the most important supervised learning algorithms are:

- k-Nearest Neighbours
- Linear Regression
- Logistic Regression
- Support Vector Machines (SVMs)
- Decision Trees and Random Forests
- Neural networks

Certain neural network architectures can be unsupervised (auto-encoders, restricted Boltzmann) or semi-supervised (deep belief networks, unsupervised pre-training) (Géron, 2017). Figure 2.1 below illustrates how supervised machine learning works on a conceptual level.

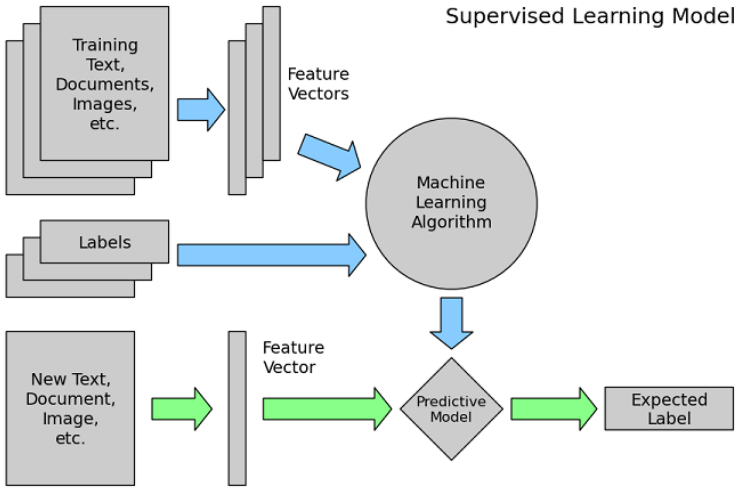



Figure 2.1: Supervised learning conceptual diagram, from Scikit-learn (2011).

The first step of supervised learning is to divide the data into two different sets: a training set and a test set. The machine learning model will be created with the training set, and the test set is used to evaluate the performance of the machine learning model. Training data is transformed into feature vectors so that the machine learning algorithm can fit this together with the accompanying label data into a predictive model. The final predictive model is then able to make expected labels for the test set, which can then be compared to the actual labels (ground proof) to estimate how well the model performs.

### 2.2.3 *Support Vector Machines*

A Support Vector Machine (SVM) is a popular Machine Learning model that is particularly well suited for for classification of complex but small- or medium-sized datasets (Géron, 2017). It is capable of performing linear, nonlinear classification, regression, and outlier detection. The main concept behind SVM classification is to separate classes by fitting the widest possible 'street' (or margin width) between classes. Figure 2.2 visualizes this widest street classification. In the figure red and green instances are separated by the dotted line. This line is established by calculating the largest margin width. The margin lines that run parallel with the decision boundary are determined by the instances located on these lines. These instances are called the support vectors. Any instance that is not on the "street" is not a support vector, and has no influence on the decision boundary. Computing predictions in this model is therefore only based on the support vectors, and not the whole training data set.

Defining a strict model where all instances are outside of the boundary margin and where each instance is positioned on the correct side of the boundary is called hard margin classification. Two consequences of hard margin classification are that the model can only be applied linearly separable data, and that outliers can influence performance a lot. A more flexible approach is soft margin classification. Here the model tries to compromise between completely separating classes and having the widest margin or

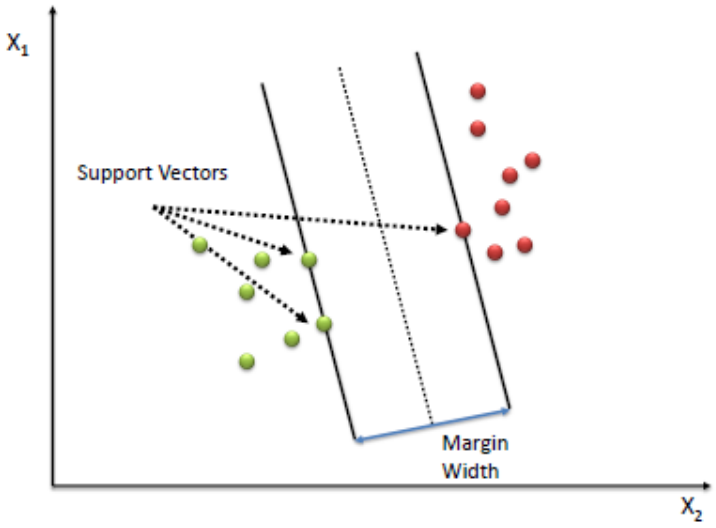



Figure 2.2: Support Vector Machine algorithm visualization from Sayad, Saed (2012).

street. This is defined through a hyper-parameter that allows for tuning between margin width and allowance of margin violations. SVM on non-linear data can be approached with the following methods: polynomial features, polynomial kernel, adding similarity features, or using an RBF kernel (Géron, 2017). SVM also can be applied to a regression task. The trick is to reverse the objective: instead of trying to fit the largest possible street between two classes while limiting margin violations, SVM Regression tries to fit as many instances as possible on the street while limiting margin violations (Géron, 2017)

### 2.2.4 *K-Nearest Neighbours*

K-nearest neighbor (KNN) is a very simple machine learning algorithm in which each observation is predicted based on its 'similarity' to other observations (Boehmke and Greenwell, 2019). The algorithm stores all available data points and calculates the distance between observations in a feature space. Commonly used distance calculation metrics are Euclidean, Manhattan, and Minkowski distance (Cunningham and Delany, 2007) The K stands for the amount of nearest neighbors specified by the user, so if K=5 the algorithm will find the five nearest observations. KNN can be applied to classification tasks as well as regression problems. The main difference in application is that with classification a majority voting takes place, while with regression a mean is calculated between points. Although KNN usually isn't the best choice in terms of performance, it does not require a lot of parameter tuning to perform reasonably well. It also handles non-linear relationships without any data engineering steps.

### 2.2.5 *Ensemble Learning*

Ensemble learning is a machine learning approach that combines several predictors: predictions of all models are aggregated, after which a majority voting takes place. Quite often this results in a better prediction than with

the best individual predictor (Géron, 2017). A group of predictors is called an ensemble, which is why the technique is called Ensemble Learning, and a specific Ensemble algorithm is called an Ensemble method (Géron, 2017). One widely used ensemble method is tree-based learning, which utilizes ensembles of decision trees to predict. Figure 2.3 provides an overview of how tree-based learning has evolved over the years, which starts with decision trees and ends with optimized Gradient Boosting.

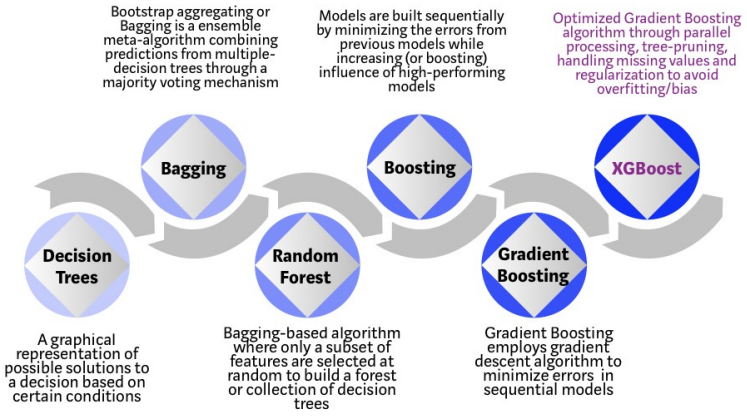



Figure 2.3: Evolution of tree-based learning overview, from Morde, Vishal (2019).

*Decision Trees*

Decision trees work by breaking down a dataset into smaller subsets recursively, ultimately forming a tree structured that can be understood and traversed. The tree structure consists of decision nodes that lead into leaf nodes. A decision node has two or more branches, with each containing a value for the tested attribute (True/False; value within range, etc). The leaf node is the result after traversing the decision tree. Decision Trees make very few assumptions about the training data (Géron, 2017), which means that without regularization steps the decision tree model will fit itself around the dataset. This results in overfitting of the model and should be prevented.

*Random Forests*

A Random Forest model consists of an ensemble of decision trees, and uses a data sampling method called bagging. Bagging (which stands for Bootstrap Aggregating) is a general-purpose process for reducing the variance of a statistical learning method (James et al., 2013), which is quite useful when applied to decision trees, since these are prone to overfitting (James et al., 2013). In bagging, multiple predictors (decision trees) of the same machine learning algorithm are trained on different subsets of the training data. When the predictors are trained, the ensemble predicts a final value by aggregating the predictions of all predictors. Aggregation for regression is done by calculating the average between all predictors, and for classification majority vote is picked (also called mode). Each individual predictor has a higher bias than if it were trained on the full training set, but aggregation reduces both bias and variance (Géron, 2017). This leads to a final model with similar bias but a lower variance (less overfitting) than a

single decision tree trained on the full data set. One disadvantage of this approach however is that the model becomes more difficult to interpret (James et al., 2013). Random Forests provide an improvement over bagged trees by tweaking the algorithm so the generated trees become decorrelated (James et al., 2013). Extra randomness is introduced when growing trees; instead of searching for the very best feature when splitting a node, the algorithm searches for the best feature among a random subset of features. This makes it so the overall strongest feature isn't always picked first, and gives other (moderately strong) features more of a chance. The result of this is a more diverse (decorrelated) set of decision trees, which results in an overall better predictive model (Géron, 2017).

*Boosting*

Boosting is an approach that is similar to bagging and Random Forest (James et al., 2013) in that it uses multiple weak learners (in this case decision trees) and combines them into a strong learner. The key difference however is that with Boosting the decision trees are created sequentially: each subsequent tree is created using information from previously grown trees. This is different from bagging because there is no bootstrap sampling of the data involved. Instead, the decision tree is trained on a modified version of the complete dataset. Figure 2.4 provides a visual comparison of boosting, bagging, and single iteration (normal) machine learning. Generally, statistical learning approaches that learn slowly tend to perform well (James et al., 2013). In Boosting this is done with the sequential fitting of the decision trees. First a decision tree is trained on the data set. The next decision tree is fit using the residuals from the earlier decision tree. This step is repeated a number of times with updated residual data as input. Each tree can be small with only a few nodes.

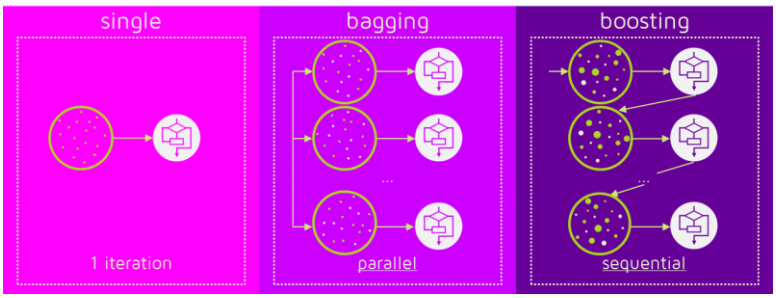



Figure 2.4: The difference between bagging and boosting, from aporras (2016).

There are many different boosting methods available, with the most popular being AdaBoost (Adaptive boosting) and Gradient boosting (Géron, 2017). The AdaBoost method was originally proposed by Freund et al. (1999) and works by assigning weights to incorrectly classified observations. Gradient boosting on the other hand utilizes a technique called Gradient Descent. XGBoost (eXtreme Gradient Boosting) by Chen and Guestrin (2016) is a widely used implementation of gradient boosting that is capable of achieving state-of-the-art results.

## 2.3 MACHINE LEARNING APPLIED IN GEOGRAPHY-RELATED FIELDS

There is not much literature on the applications of machine learning in the field geography currently available. The reason for this is that advancements in Machine Learning have occurred only relatively recently, which means that spillover of methods to other fields of study is only just beginning. Much of the foundation of ML in Geography has yet to be formed, best practices and state-of-the-art approaches to prediction tasks still need to be found. For example, Brunsdon (2016) provides a progress report on reproducible quantitative research in human geography, but does not mention any applications of artificial intelligence, machine learning or data science. However, he notes that trends suggest a turn towards "the creation of algorithms and codes for simulation and the analysis of Big Data", which indicates a willingness within the field to move towards a Geographic Data Science (outlined in Singleton and Arribas-Bel, 2019), but is simply not at that stage yet. Additionally, while some AI implementations do exist within geography (Hu et al., 2019), this is predominantly for physical geography and not so much human geography. These are applications such as: automatic terrain feature recognition, land cover classification, and ecological habitat prediction.

A relevant related field to acquire potential domain knowledge from is that of real estate. The relevance to gentrification research is based on the fact that multiple measures of neighborhood gentrification are at least partly derived from housing prices (Reades et al., 2019; Holm and Schulz, 2018; Guerrieri et al., 2013). The assumption that we make is that there exists enough similarity between these two types of research, in terms of data and on a conceptual level, that domain knowledge for real estate value prediction (i.e. model specification, parameter tuning, dimension reduction) is potentially useful for gentrification prediction as well. Graczyk et al. (2010) compare bagging, boosting, and stacking ensembles applied to real estate appraisal. Their results show that there is no single algorithm which produces the best ensembles. Park and Bae (2015) utilize and compare a selection of machine learning models for housing price prediction of Fairfax County, Virginia. The selected algorithms are C4.5, RIPPER, Naive Bayes, AdaBoost; C4.5 and RIPPER are decision based. Performance is measured via minimum error rate, and they find that RIPPER performs best, followed by AdaBoost. Wang et al. (2014) apply particle swarm optimization (PSO) in combination with Support Vector Machine (SVM) to real estate price forecasting, where PSO is used to optimize SVM parameters. Results indicate that this approach produces good real estate price forecasting performance.

## 2.4 PRODUCTION OF GEOGRAPHIC KNOWLEDGE WITH ML

Singleton and Arribas-Bel (2019) formulate two important considerations when it comes to Data Science applied to geographic questions: 1) of what or where are the underlying data representative, 2) how divergent is the extraction of knowledge within this context from more widely accepted epistemologies such as those emerging from Quantitative Geography, Geographic Information Science, or Geocomputation?

Supervised machine learning uses labeled data to learn and serves as the ground truth for the predictive model. This label can be a category (nominal), point scale (ordinal), or an exact value (ratio). In traditional machine learning prediction tasks the assumption that a label for a certain case ac-

curately reflects truth is generally accepted. Usually the labeling task itself is relatively straight-forward, and measuring of inter-annotator agreement is quite often used to obtain a robust set of labelled data. Inter-annotator agreement is a measurement score used to assess the reliability of an annotation process, which is a requirement if we want to assume that our dataset and subsequent analysis are methodically correct. The most common way of reporting agreement is via Cohen's Kappa, Fleiss K, or Krippendorff's Alpha (Artstein, 2017). When it comes to geographic research this labelling approach becomes much more of a challenge. The primary reason for this difficulty is that the subject of prediction —gentrification in this case— is a lot more complex in nature. For example: we can establish relatively easily when an image contains a car or not. Gentrification, on the other hand, is such a broad and multi-faceted concept that it is difficult to define and label.

Ultimately, the question arises how closely it is that our data and predictive model approximate reality. In other words, are the results from our machine learning model sufficiently representative? When we attempt to define when data and method are sufficiently representative, we need to take into consideration correctness as well as feasibility of our method. There is a trade-off between quality and quantity of research. If requirement definitions for data quality are too strict, research will never be good enough. On the other if we are too pragmatic in our approach, results lose in value. Trade-off decisions need to be documented, so we can take into account considerations when compromising on data and approach. In terms of predicting gentrification this means it is important to explain what data is used, how it is used, and what potential limitations of the method are. Which definition of gentrification are we attempting to quantify, and what relevant aspects are we not capturing with our data that could significantly influence our results? Geography should not be reduced to a set of spatial coordinates in a machine learning data set, because this strongly increases the risk of not being able to adequately assess the value created predictive machine learning model. Data can be interpreted free of context and domain-specific expertise, but the result will be that epistemological interpretation is likely to be anaemic or unhelpful as it lacks embedding in wider debates and knowledge Kitchin (2014). A new Data-driven approach of doing research should not forgo domain expertise, but instead be integrated within the research space of geography. Geography has the potential to complement Data Science by bringing, literally and epistemologically speaking, the role of context and decades of experience with these questions (Singleton and Arribas-Bel, 2019). If we want self learning systems to advise and aid us in answering scientific questions we also be able to critically evaluate this new approach in order for Machine Learning to useful in solving geographic challenges. This is an important role for geographers. Kitchin (2014) suggests a new epistemology that employs the methodological approach of data-driven science within a different epistemological framing that enables social scientists to draw valuable insights from Big Data that are situated and reflexive.

# 3 METHOD

The aim of this thesis is to analyze and evaluate the effectiveness of machine learning applied in geographic context. This is done via the task of neighborhood gentrification prediction as outlined in Reades et al. (2019). The analysis consists of three parts. Firstly, comparing performance between a selected set of machine learning algorithms in order to find out which model approach is best suited for the task of predicting gentrification in terms of regression evaluation metrics. Secondly, the goal is to find out which variables/features are most important in explaining each machine learning model. Model explanation is done via the implementation of the SHAP (Lundberg and Lee, 2017) library. SHAP stands for SHapley Additive exPlanations. The third part of this research is to forecast and compare future gentrification, using the best performing models.

## 3.1 DATA

The original data used comes from research presented by Reades et al. (2019), and consists of processed census data from the 2001 and 2011 UK Census of Population and the London Data Store [1]. Table 3.1 contains the variables that are used to construct Socio-economic scores (SES) for London neighborhoods. These variables encompass neighborhood averages for income, house value, occupational employment, and qualification level. Table 3.2 contains the variables used to predict the corresponding SES for London neighborhood level gentrification.

Table 3.1: Gentrification composite score (SES) variables

| London Scoring Data |
|---|
| LSOA Household income |
| Median housing & sales |
| Occupational share |
| Highest level of qualification |

Table 3.2: Modeling data used to predict London gentrification

| London modeling data | |
|---|---|
| Green space & access | Age structure |
| Dwelling period built | National Socio econ classification (NS-SeC) |
| Travel time to major infrastructure | Economic activity |
| Travel time to bank station | Country of Birth |
| My Fare Zone | Dependent children |
| Travel mode | Population density |
| Cars & vans | Household composition |
| Real Estate tenure | Industry |
| Hours worked | Marital status |
| Ethnicity | Religion |

[1] Data, processing scripts and further explanation are available at https://github.com/jreades/urb-studies-predicting-gentrification.

## 3.2 METHOD ANALYSIS

Neighborhood socio-economic score (SES) is predicted with the following machine learning regression algorithms:

- Support Vector Regression (SVR)
- K Nearest Neighbor (KNN)
- Random Forest (RF)
- XGBoost
- CatBoost
- AdaBoost
- Linear Regression
- Ridge Regression

Reades et al. (2019) use Random Forests to analyze and predict gentrification in London neighborhoods, based on 2001 and 2011 Census variable data. Gentrification at the neighborhood level is operationalized in term of socioeconomic score (SES), which is calculated through a combination of variables: household income, house value, occupational share, and highest level of qualification at neighborhood level. Principal Component Analysis (PCA) is used to obtain the SES, which is used to measure neighborhood ascent or descent. Model prediction performance is analyzed and evaluated by training the model on 2001 data and testing this on 2011 actual values. Data from 2011 is then used to predict those areas most likely to demonstrate 'uplift' or 'decline' by 2021. The results show improvement over linear regression even without hyperparameter tuning.

The machine learning models are trained with census data from 2001 to predict SES Ascent target scores (2011 SES minus 2001 SES). To predict future gentrification, the trained model is given 2011 census data. The technical analysis is done in Python. Python is a high-level programming language suited for scientific and engineering code that for most cases is fast enough to be immediately useful as well as flexible enough to be sped up with additional extensions (Oliphant, 2007). Machine learning algorithms implemented via Scikit-learn (Pedregosa et al., 2011), which is a Python module that integrates a wide range of state-of-the-art machine learning algorithms for medium-scale supervised and unsupervised problems (Pedregosa et al., 2011). Data processing and structuring is done primarily with the Pandas library (McKinney, 2011).

## 3.3 ALGORITHM PERFORMANCE

### 3.3.1 *Evaluation metrics*

Since the choice has been made to perform a regression task, the following metrics will be used to evaluate and compare prediction performance between models: $R^2$, Mean Squared Error (MSE), Mean Absolute Error (MAE), and explained variance. The $R^2$ takes on a value between 0 and 1 and measures the proportion of variability in Y (dependent variable) that can be explained using X (independent variable) (James et al., 2013). MSE measures the average of error squares, which is the average of squared differences

between predicted values and true (expected) value. MAE measures the average magnitude of errors without considering direction. These evaluation metrics are decided upon in order to stay consistent with the earlier results by Reades et al. (2019).

### 3.3.2 *Optimization*

Gentrification prediction performance of the different models is optimized with hyper-parameter tuning. Hyper-parameters are parameters that are not directly learnt within estimators, instead they must be set prior to training and remain constant during training of the model. Tuning hyperparameters is an important part of building a Machine Learning system (Géron, 2017). Hyper-parameter tuning is normally carried out by hand, progressively refining a grid over the hyperparameter space (Bardenet et al., 2013). For this thesis the Scikit-Learn module GridSearchCV is used to exhaustively consider sets of user specified parameter combinations.

## 3.4 MODEL INTERPRETATION

An important aspect of obtaining new knowledge and understanding in scientific research is to find out why phenomena happen in the way that they do, and is usually done via the interpretation of results. In statistical methods like linear regression this is achieved by interpreting coefficients, but unfortunately in machine learning there is little consensus on what interpretability is, and how to evaluate machine learning models for benchmarking (Doshi-Velez and Kim, 2017). These relatively complex machine learning models often have a more accurate predictive capability, but as a downside their complexity means they are regarded as black-boxes when it comes down to interpretation. The easy way to circumvent this interpretation problem would be to just use an appropriate linear model instead (Linear/Ridge), but with the increasing availability of Big Data (Kitchin, 2014) this problem becomes worthwhile to solve for geographic research. The reason for this is that machine learning approaches produce better results when it comes to Big Data. Lundberg and Lee (2017) propose a framework called SHAP (SHapley Additive exPlanations) to solve this lack of interpretability in Machine Learning. SHAP is a unified approach to interpreting model predictions, and utilizes game theory to accomplish this. The SHAP framework implementation is available as a library in Python and R programming languages.

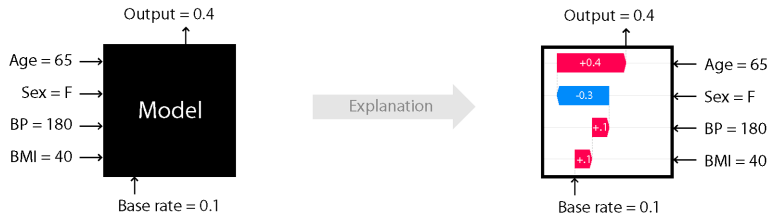



Figure 3.1: A blackbox model versus SHAP conceptual visualization, from Lundberg et al. (2019).

The approach of SHAP to explaining complex models such as ensemble methods or deep learning models is to not use the original model, but

rather to define a simpler explanation model which is an interpretable approximation of the original model. This is done via a method called Additive Feature Attribution, whereby the explanation model is a linear function of binary variables (Lundberg and Lee, 2017). SHAP functions primarily a local method, which means that explanation is performed on instance level. Global interpretation is however also possible. This is achieved by aggregating the Shapley values derived from local interpretation. Lundberg and Lee (2017) mathematically define Additive Feature Attribution in the following way:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i.$$

In this function, $g$ is the explanation model, $z' \in \{0,1\}^M$ is the coalition vector of simplified features, $M$ is the number of simplified input features, and $\phi_i \in \mathbb{R}$ is the feature attribution for a feature $i$. In the coalition vector, an entry of 1 means that the corresponding feature value is "present" and 0 that it is "absent" (Molnar, 2019). Explanation model $g$ uses simplified inputs $x'$ derived from the original model $f(x)$, where the simplified inputs map to the original inputs through a mapping function $x = h_x(x')$. Local methods try to ensure $g(z') \approx f(h_x(z'))$ whenever $z' \approx x'$. Simply put: the explanation model $g$ should accurately represent the original black box model $f$.

SHAP Additive Feature Attribution utilizes a concept from coalitional game theory called Shapley values. Shapley values in game theory are about fairly allocating credit to player contributions in a game. Shapley in Machine Learning focus on fairly allocating credit to features as they come into a model. These features can potentially contribute unequally on the output of a model, depending on the feature order. The way in which Shapley values are applied to explaining of Machine Learning models is by comparing a machine learning prediction to a game's payout and to see each feature in the prediction model as a contributing player in a coalition. Certain players contribute more to the payout than other players, and Shapley values allow us to quantify this contribution. A Shapley value in Machine Learning is therefore defined as "the average marginal contribution of a feature value across all possible coalitions" (Molnar, 2019). It is good to keep in mind that the Shapley value is NOT the difference in prediction when we would remove the feature from the model (Molnar, 2019). Strong interaction effects can exist between features so we should not pick a particular order and assume this sufficiently captures the phenomenon. In order to account for these interaction effects, Lundberg and Lee (2017) define three desirable properties for SHAP as axioms of fairness. These properties are 1) Local accuracy, 2) Consistency, and 3) Missingness. Local accuracy (also called additivity) is when the sum of the local feature attributions equals the difference between the base rate and the model output. The credit allocation sum from the expected model has to be equal to the actual model output. Simply put: Shapley credit has to be fully allocated, there is no extra or left over credit.

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i.$$

The explanation model $g(x')$ matches the original model $f(x)$ when $x = h_x(x')$ , where $\phi_0 = f(h_x(0))$ represents the model output with all simplified inputs toggled off (i.e. missing). Consistency (also called monotonicity): If you change the original model such that a feature has a larger impact in every possible ordering, then that input's attribution should not decrease. Example: if we have two models where in one model a certain feature has greater impact regardless of feature ordering, then the credit given to that same feature in the lesser impact model should never have a higher value. Violating consistency means you can't trust the feature orderings based on your attributions, even if it's within the same model.

The third property is missingness. If the simplified inputs represent feature presence, then missing- ness requires features missing in the original input to have no impact (Lundberg and Lee, 2017). The Missingness property enforces that missing features get a Shapley value of 0. In practice this is only relevant for features that are constant. Missingness says that a missing feature gets an attribution of zero. Note that $x'_j$ refers to the coalitions, where a value of 0 represents the absence of a feature value. In coalition notation, all feature values $x'_j$ of the instance to be explained should be '1'. The presence of a 0 would mean that the feature value is missing for the instance of interest. Mathematically this is written as:

$$x'_j = 0 => \phi j = 0$$

There exist multiple different implementations of Additive Feature Attribution, notable ones being LIME (Ribeiro et al., 2016), DeepLIFT (Shrikumar et al., 2016), and Classic Shapley Value Estimation ( Datta et al., 2016; Lipovetsky and Conklin, 2001; Štrumbelj and Kononenko, 2014). SHAP attempts to unify these implementations into a framework by incorporating SHAP values. Global Shapley values result from averaging over all N! possible orderings.

For this thesis we use the following two SHAP implementations:

- Kernel SHAP: used to explain the output of any function. Kernel SHAP uses a special weighted linear regression to compute the importance of each feature. Kernel SHAP combines the ideas of Shapley values with a linear feature attribution method called LIME.
- Tree SHAP: used to explain ensemble tree models. Tree SHAP is a variant of SHAP specifically made for tree-based machine learning models, such as random forests, decision trees, and gradient boosted trees (Lundberg et al., 2019). An advantage that Tree SHAP has over Kernel SHAP is that in terms of implementation computational complexity is reduced. This makes it perform the analysis much faster than Kernel SHAP, which makes it more viable to use in a practical sense.

The idea behind SHAP feature importance is simple: Features with large absolute Shapley values are important. Since we want the global importance, we average the absolute Shapley values per feature across the data (Molnar, 2019).

## 3.5 GENTRIFICATION RESULT ANALYSIS

In order to explore the value that machine learning can provide to understanding gentrification and also in a broader context to geography, the pre-

dictive output of future gentrification will be analysed via GIS visualization. The machine learning results analysis will be performed in following way:

- Spatial pattern from model predictions
- Differences and similarities in prediction visualizations between machine learning models

Gentrification itself is measured via socioeconomic status (SES), which is derived from LSOA Household income, Median housing, Occupational share, and Highest level of qualification with the use of Principle Component Analysis (PCA). From this score we can ascertain whether a neighborhood's score increases (SES Ascent) or decreases (SES Descent). Future prediction means we will be able to forecast which neighborhoods are expected to be undergoing gentrification. Prediction scoring can be performed in two ways: absolute value change, and proportional change via a rank-based calculation. Ranking of neighborhoods allows us to compare neighborhoods to each other directly, and prevents the overall increase of housing market prices to influence the results. The observation of relative changes is important for our analysis, because low-status neighborhoods might not look like they are undergoing gentrification in terms of absolute numbers, which mean might incorrectly denote them as not-gentrifying. The opposite might also be true; affluent neighborhoods can fluctuate more easily when measured in absolute numbers.

In order to evaluate the performance and accuracy of our predictive models, data is required of what is being predicted. This is done by inputting feature data from 2001 and then comparing the predicted output with data from 2011. However, when the goal becomes to predict future gentrification using current data this evaluation is obviously not possible until such data becomes available in the future. As such, we will instead compare similarities and differences of future prediction visualizations. The comparison of different predictive models can help us in evaluating the robustness of machine learning in geography. Different algorithms possibly predict vastly different spatial trends, which raises questions pertaining to theoretical underpinnings (why are results different, what is truth, if we have to choose one model which one should it be) and implications for planning policy. This could also provide new insight in terms of future machine learning model building and feature selection. Which features have a strong influence on future prediction, and how might the combination of data and algorithm introduce bias to the obtained results?

# 4 RESULTS AND DISCUSSION

In this chapter we will present an overview of the results from the gentrification and SHAP analysis, and attempt to answer the research questions of this thesis formulated in the first chapter. Additionally, the analysis results will be used to discuss and evaluate upon the potential role of Machine Learning within the field of Geography.

## 4.1 RESULTS

### 4.1.1 *Algorithm performance*

Table 4.1 contains the performance results of different machine learning models on neighborhood gentrification prediction in London. The evaluated models have all been optimized: this means that algorithm hyperparameters have been tuned to produce the best performing predictive model. Hyper-parameter tuning was done partly via Scikit-learn *GridsearchCV()* module and partly via manual testing. Prior results from Reades et al. (2019), as well as a Linear Regression have been included as a benchmark. From table 4.1 we observe that the best performing algorithm is XGBoost with an $R^2$ of 0.707, and is slightly better than the best trained model from Reades et al. (2019). The KNN and SVR models do not outperform the optimized Random Forest, however they are still slightly better than a linear regression approach. The AdaBoost algorithm appears to be the least suited for this regression task and performs the lowest at an $R^2$ of 0.563. Table 4.2 contains the execution times of the used machine learning models. In these times are included training and testing of the model. In the table we observe that Linear regression is the fastest to finish the run at 0.045 seconds, which is not unexpected due to it being a relatively straightforward method. Random Forest regression on the other hand takes much longer to run with a runtime of 22.861 seconds, making it the slowest method in our list of approaches. This is due to the Random Forest algorithm requiring relatively complex computations, which increases the execution time. Table A.2 lists the setting used.

Table 4.1: Performance results of optimized models

| Model | R2 | MSE | MAE | Expl. Var. |
|---|---|---|---|---|
| XGBoost | 0.70722 | 0.18176 | 0.27691 | 0.70867 |
| Random Forest (Reades et al., 2019) | 0.69825 | 0.18733 | 0.25944 | 0.70184 |
| KNN | 0.65519 | 0.21406 | 0.28298 | 0.65780 |
| SVR | 0.648 | 0.219 | 0.274 | 0.649 |
| Linear regression | 0.63980 | 0.22362 | 0.30430 | 0.64071 |
| AdaBoost | 0.56306 | 0.27126 | 0.34756 | 0.59379 |
| Ridge Regression | 0.64054 | 0.22316 | 0.30458 | 0.64141 |
| CatBoost | 0.69696 | 0.18814 | 0.26368 | 0.69931 |

Table 4.2: Model runtime

| Algorithm | Runtime (in seconds) |
|---|---|
| XGBoost | 3.068s |
| Random Forest | 22.861s |
| AdaBoost | 5.446s |
| KNN | 0.946s |
| SVR | 2.529s |
| Linear regression | 0.045s |
| Ridge Regression | 0.029s |
| CatBoost | 15.313s |

### 4.1.2 *Feature importance*

This subsection contains the results from the SHAP analysis for each tuned algorithm, and is used to ascertain which features from our data are important for the machine learning model. The idea behind SHAP feature importance is straightforward: features are considered important when they have a large absolute Shapley value. This importance is established for individual predictions, but for this experiment we want to look at global importance of features in a machine learning model. To obtain this global importance, we take the average of all absolute Shapley values across the data for each feature. We then rank the features based on these averaged Shapley values to get an overview of global importance. In the case of figure 4.1 the most important features are House Prices, Household Income, and Males:49 or more hours (variable for the amount of males working 49 or more hours in a week).

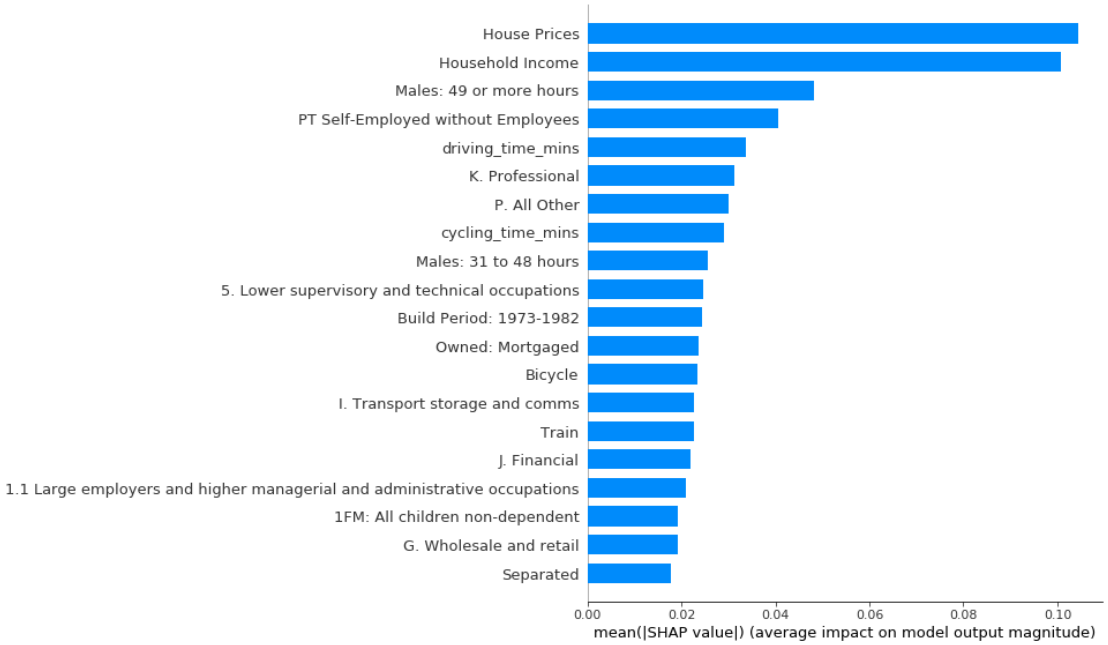



Figure 4.1: SHAP Global Feature Importance plot of gentrification prediction with XGBoost .

Figure 4.2 is a SHAP summary plot that visualizes global feature importance for the XGBoost gentrification prediction model and combines it with

feature effects. In the figure a set of top 20 features is ranked from having most impact to having least impact on model prediction. The plot itself is made up of many points; each point represents a Shapley value for a feature and an instance. The positions of these points are determined by the following characteristics:

- Vertical location shows what feature it is depicting
- Color shows whether that feature was high or low for that row of the dataset
- Horizontal location shows whether the effect of that value caused a higher or lower prediction.

Overlapping points are jittered in y-axis direction: this allows us to get a sense of Shapley value distribution on a per feature basis. For example: in figure 4.2 for the feature "House Prices" we can see that a lot of instances with a low feature value (colored blue) have a very similar negative impact on model output. The instance colors make it possible to infer patterns between feature values and model impact. For the feature "House Prices" this means that high house prices have an overall moderate to strong positive impact on model output, and low house prices have a minor negative impact on model output. Other features such as 'driving_time_mins' have a reverse relationship between feature values and SHAP value, whereby high feature value results in slight negative and low feature value in positive impact.

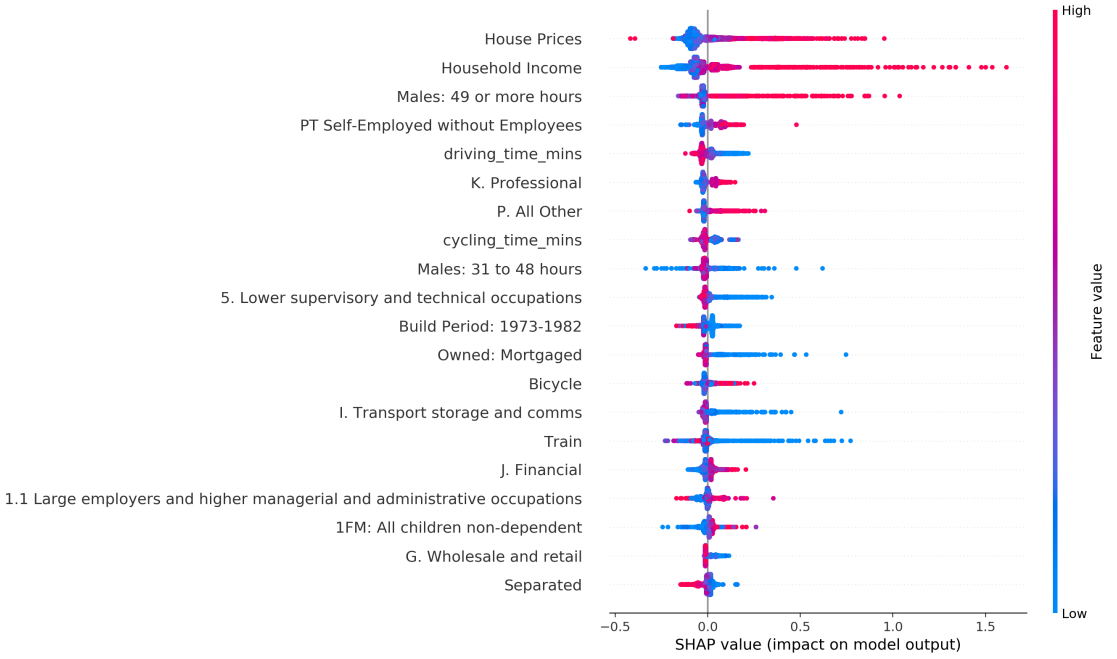



Figure 4.2: SHAP summary plot for XGBoost that takes into account intensity and direction of feature impact.

Figure 4.3 shows the SHAP summary plot for the Random Forest Gentrification prediction model. When we compare this plot with the XGBoost summary plot we see that in terms of global feature importance both models rely on the same top three features: House Prices, Household Income, and Males: 49 or more hours. All the other important features are different

between the two predictive models. Also, if we look more closely at the dot patterns of the top three important features we observe that distribution of SHAP values, although similar, is not the same for the two summary plots. Global feature impact is the same, but in terms of specific prediction there is variation. This indicates that different algorithms find different types of data important for prediction. Since both algorithms have a similar overall performance (see 4.1), it is useful to understand how they deviate in terms of decision-making . Although it is not a revelation to know that two different algorithms function in a different way, the way in which they incorporate feature data can be of relevance when we try to model and understand phenomena such as gentrification. In terms of data availability this can play a role, as well as with interpretation and application. Knowing why the algorithm does something is an important thing to consider, also because it raises the question of which predictive we should then trust. There are considerable implications for gentrification forecasting in an applied setting, for example when such a system is used to assist with policy decision-making.

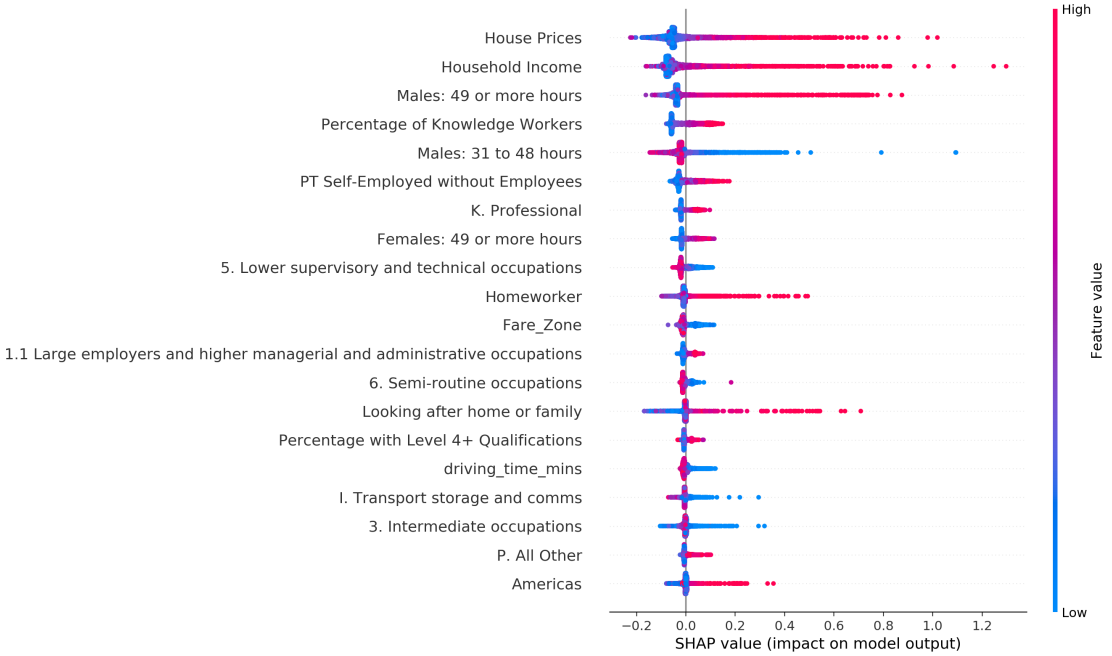



Figure 4.3: SHAP summary plot for Random Forest that takes into account intensity and direction of feature impact.

Figure 4.4 shows us the SHAP summary plot results for Ridge regression, which is a variation of OLS Linear regression. These results are included because they allow us to compare our complex machine learning best approaches (XGBoost and Random Forest) with a relatively straightforward regression. Considering the performance of Ridge regression for SES prediction (see table 4.1) is quite good, this makes it a good benchmark candidate. In the summary plot we see that the Ridge regression model depends on different features than the previous two models. The top three features with largest impact on model output are Household Income, Higher professional occupations (NS-SeC), and Black (ethnicity). What this means is that depending on the approach (algorithm) you will find distinct variables or features that will need to be addressed once this gets used in practical setting (e.g, policy-making). The fact that this model has determined that

a feature such as ethnicity (in this case Black) negatively impacts socioeconomic score presents us with very important (political) considerations.

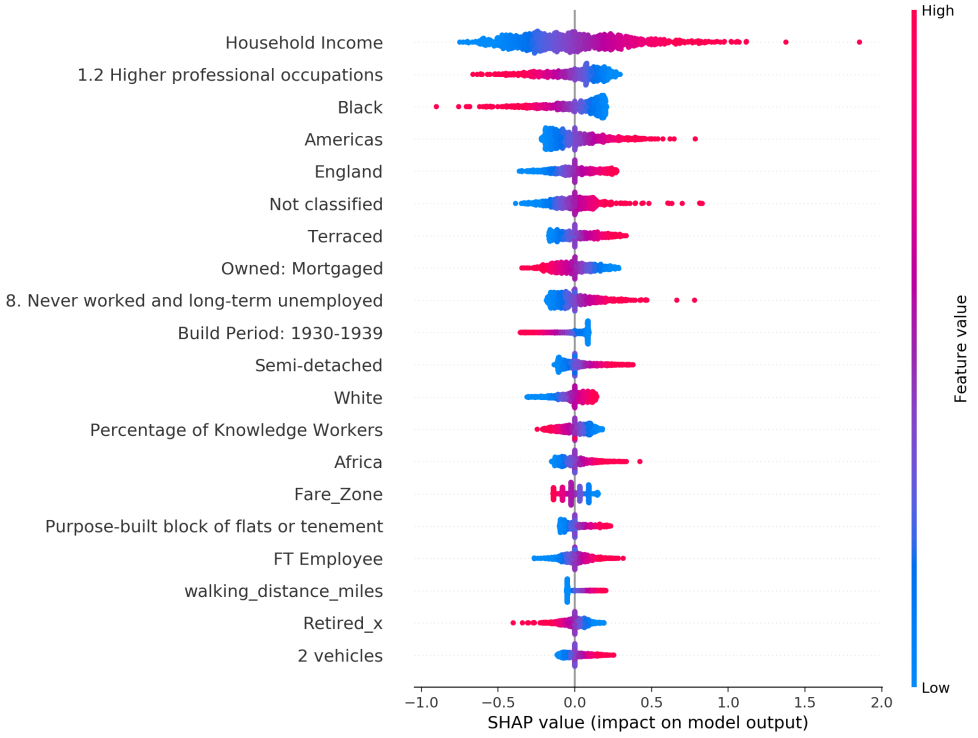



Figure 4.4: SHAP summary plot for Ridge regression that takes into account intensity and direction of feature impact.

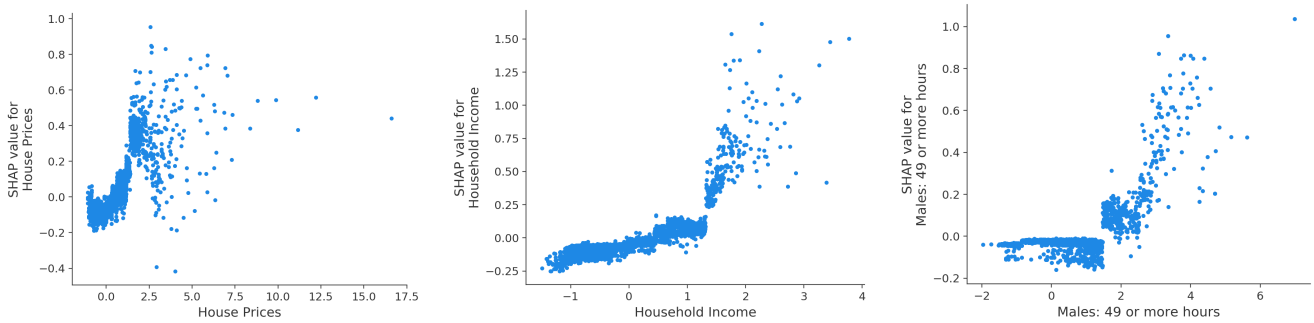
The summary plots allow us to observe first indications of the relationship between the value of a feature and the impact on the prediction, but in order to see the exact form of the relationship, we will have to look at SHAP dependence plots. A dependence plot is a scatter plot that shows the effect a single feature has on the predictions made by the model.

Figures 4.5a, 4.5b, and 4.5c show the dependence plots for the top three most important features in the XGBoost model. Each dot in the scatter plot is a single prediction from the dataset. The X-axis is the value of the feature, and the Y-axis is the SHAP value for that feature. The SHAP value represents how much knowing that feature's value changes the output of the model for that sample's prediction. Due to normalization of the data via unit variance scaling and mean centering, feature values on the X-axis are not easy to interpret directly. The dependence plots however do allow us to interpret the nature of the relationship between feature values and predictive model impact. Here we see that the relationship between feature value and model importance is not linear. A general observation is that the higher a feature value is, the more scattered the dots become, indicating that model impact is not uniform at the higher end of the plot. We do see that overall a positive correlation is present for the features 4.5a, 4.5b, and 4.5c. A higher feature value seems to indicate a positive SHAP value, which means that it has a positive impact on gentrification prediction. In other words: the higher a neighborhood's household income, house prices, and amount of males working 49 or more hours, the more likely that neighborhood is to undergo gentrification in the (near) future according to our predictive

model. A technical observation from the plots is that we can identify a cut-off in the scatterplot patterns, especially apparent in figures 4.5b, and 4.5c. This is possibly due to the XGBoost algorithm defining a certain threshold in predictions.

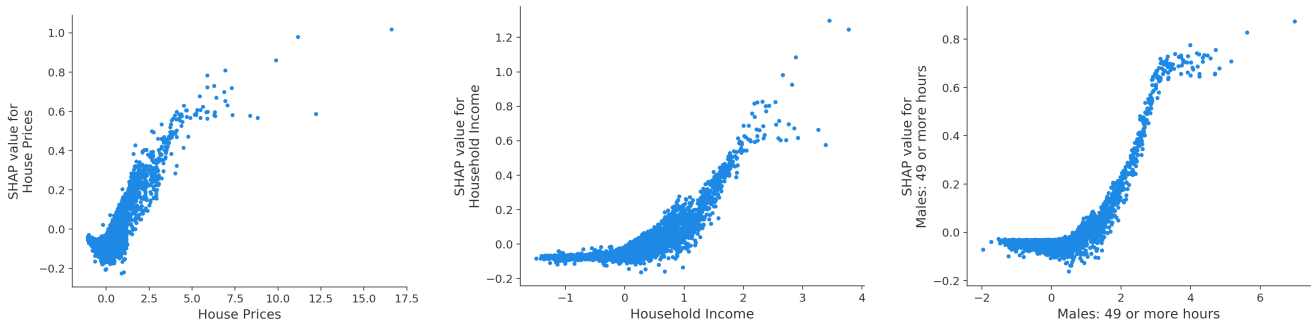
Figures 4.6a, 4.6b, and 4.6c show the top 3 model impact features for the Random Forest Algorithm, which are the same features as for the XGBoost predictive model. For both algorithms the top 3 features are the same (House Prices, Household Income, Males: 49 or more hours), but when we compare the dependence plot patterns we see that the way in which feature values impact model output is different. The scatter plots from figure 4.6 show a more strongly defined pattern compared to the plots from 4.5, as well as a more clustered distribution of the dots. There are less outlier dots overall, and there is no threshold cut-off pattern present. A similarity between models can be found in the direction of the correlation between feature value and SHAP value: in both XGBoost and Random Forest these are positive. The SHAP values themselves are also relatively comparable, which means that feature values have a comparable impact on model prediction magnitude.

The dependence plots for the most important features of the Ridge regression model summary plot (figure 4.4) show a completely linear correlation between feature value and SHAP value: this is to be expected, since Ridge regression is a linear method. The value of Household Income (fig. 4.7a) has a positive effect on model prediction, while the other two features have a negative correlation (figures 4.7b and 4.7c).



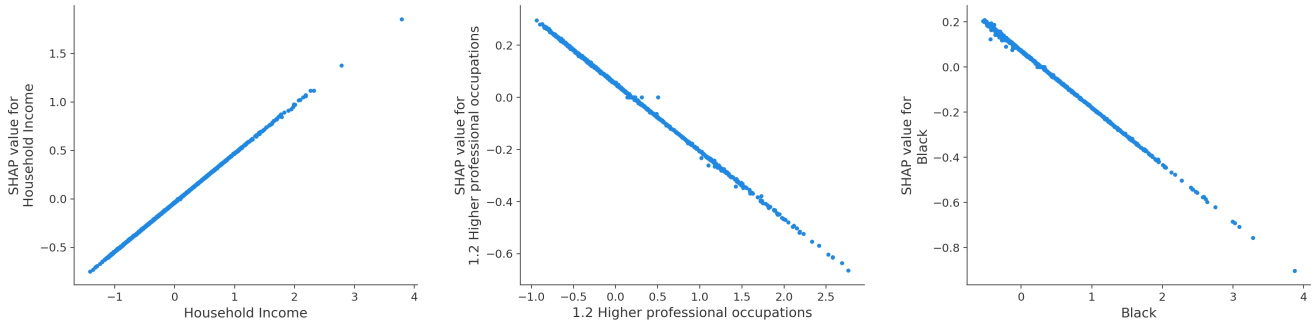

(a) Rank 1: House Prices (b) Rank 2: Household Income (c) Rank 3: Hours worked + gender

Figure 4.5: Dependence Plots: XGBoost




(a) Rank 1: House Prices (b) Rank 2: Household Income (c) Rank 3: Hours worked + gender

Figure 4.6: Dependence Plots: Random Forest




(a) Rank 1 - Household Income

(b) Rank 2 - NS-SeC: Higher professional occupations

(c) Rank 3 - Ethnicity: Black

Figure 4.7: Dependence Plots: Ridge regression

Table 4.3 lists the runtime of the SHAP explainer for each algorithm. Here we can see that SHAP calculation speed for the XGBoost algorithm outperforms every other approach, some by several magnitudes. CatBoost executes at pretty much the same runtime, and Random Forest SHAP calculation is 10x slower, albeit still quite fast. This is primarily due to the fact that these are Tree-based machine learning models, which means that they are able to use the more efficient SHAP Tree Explainer. The other algorithms have to rely on the much slower Kernel Explainer. The significance of these SHAP runtime results is that when SHAP is used for a different experiment with more data (Big Data, for example), the faster (Tree)SHAP approach will be the practical choice in this type of situation. Tree-based learning Boosting algorithms, from this perspective, appear to be the more suitable ones for Big Data tasks in geography. Due to the complex tree structure of the Random Forest and CatBoost models resulting in a long runtime, the choice was made to run TreeSHAP in approximation mode. This means TreeSHAP runs fast, but only roughly approximates the Tree SHAP values. This method only considers a single feature ordering, and therefore does not have the consistency guarantees of Shapley values and places too much weight on lower splits in the tree. The resulting approximate SHAP values do not create an issue for interpretation in this case, since in this thesis the focus mainly lies on interpretation of global feature importance. It would however have implications for interpretation if we wanted to gain a more detailed understanding of individual predictions.

Table 4.3: SHAP runtime. *= approximation of SHAP values

| Algorithm | Runtime (in seconds) | SHAP explainer |
|---|---|---|
| XGBoost | 0.713s | Tree |
| Random Forest | 8.067s | Tree* |
| AdaBoost | 405.275s | Kernel |
| KNN | 10150.900s | Kernel |
| SVR | 8604.300s | Kernel |
| Linear regression | 620.994s | Kernel |
| Ridge regression | 267.948s | Kernel |
| CatBoost | 0.855s | Tree* |

### 4.1.3 *Spatial analysis of predicted gentrification*

This section contains the visualization results of future gentrification prediction for 2021. Not all AI models have been visualized, instead we only compare the best performing models and a linear regression baseline model. This allows us to explore similarities and differences between predictive models that in terms of evaluation metrics are on par with each other, and also make it possible to see how AI models stack up to more 'traditional' quantitative linear regression. The selected AI models are: XGBoost, Random Forest, Ridge regression, and CatBoost.

The first predictive model that we look at is XGBoost. Figure 4.8 shows a histogram of the predicted SES ascent scores between 2011 and 2021 from the XGBoost model. The values represent the predicted difference in socioeconomic scores between 2011 and 2021. The mean predicted score difference is 1.361 and we also observe that the distribution is right-skewed. We also see that every predicted ascent score is positive, which means that the XGBoost model predicts a score increase for every neighborhood and no decrease. The assumption is that gentrifying neighborhoods can be identified by ascending socioeconomic scores, however to assume that a high predicted ascent equals a strong indication for gentrification might be too simple of an approach. This is because it is likely that wealthy neighborhoods will ascend faster due to a stronger increase in housing prices, and will therefore show the most drastic ascent. This will result in high-income areas being disproportionally represented in the result visualization as ascending. Gentrification is generally characterized as low-income neighborhoods becoming medium- to high-income, with resulting displacement effects. Therefore it is more useful to look at relative score increase between neighborhoods in order to capture this type of gentrification with our predictive model, which can be achieved by ranking neighborhoods. Additionally, this phenomenon of high-income neighborhoods ascending the most in our results highlights the complexity of gentrification at the conceptual level. The absolute ascent data do not represent gentrification in the standard definition, but rather it indicates the presence of a related phenomenon called super-gentrification, which Lees (2003) defines as the transformation of already gentrified, prosperous and solidly upper-middle-class neighbourhoods into much more exclusive and expensive enclaves. For our research this means that further classification of different gentrification types might be necessary for more detailed and robust interpretation of our predictive models.

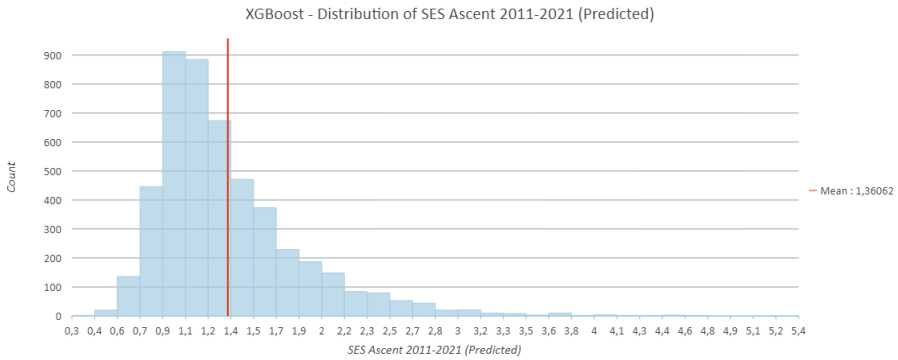



Figure 4.8: Histogram of SES Ascent prediction scores with XGBoost.

In order to somewhat correct for absolute score increase not being a very good indicator of gentrification we calculate percentile neighborhood rank change, which allows us to identify drastic ascent increase (i.e. gentrification) in a relative way. This is done by ordering neighborhoods based on SES for both 2011 and 2021, and then taking the difference between these values. Figure 4.9 shows a histogram of the standard deviation of predicted rank change for 2011-2021 with XGBoost. This makes it possible to see how many standard deviations away from the mean rank change is for each neighborhood, and allows us to identify which rank changes are significant. Z-scores are calculated by subtracting the rank change value with the mean, and dividing this by standard deviation. In figure 4.10 these standard deviation rank change values are visualized for the whole of London. The map shows only positive rank change on neighborhood level. A strong positive deviation from the average rank change suggests that the neighborhood is expected to undergo gentrification. the intensity of rank change is divided into four categories. The neighborhoods are part of a higher administrative level called a borough; label numbers and corresponding borough names can be found in the top right of the map. Figure 4.10 not does show a very clear pattern of gentrification: moderate to strong rank change appears to be predicted across the whole of London. We do observe that boroughs Barking and Dagenham (1), Newham (25), and Waltham Forest (31) contain the most neighborhoods with very strong rank change, which is more than 3 standard deviations from the mean rank change. When we look at figure 4.11, the prediction map from our Random Forest model, we see a very different spatial pattern. In this case SES rank change is much more clustered compared to our XGBoost prediction, and is more in the center of London. The most likely to gentrify boroughs are Newham (25), Tower Hamlets (30), and Hackney(12). The Ridge regression (fig. 4.12) predicts something completely different from our two other maps. The CatBoost prediction (fig. 4.13) shows spatial patterns that somewhat seem to line up with the XGBoost map, but overall does not provide use with distinct shared patterns of gentrifying neighborhoods. The main observation from comparing these different AI predictions is that there are substantial differences in which neighborhoods gentrification is expected to take place, and the predicted intensity of rank change. There appears to be no real conformity between these four predictive models, which raises the question of how we should use these predictions to gain a better understanding of future gentrification.

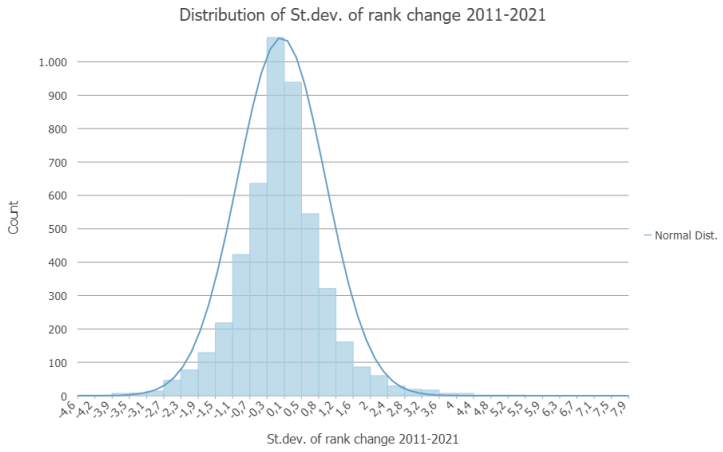



Figure 4.9: Histogram of SES Ascent prediction scores with XGBoost.

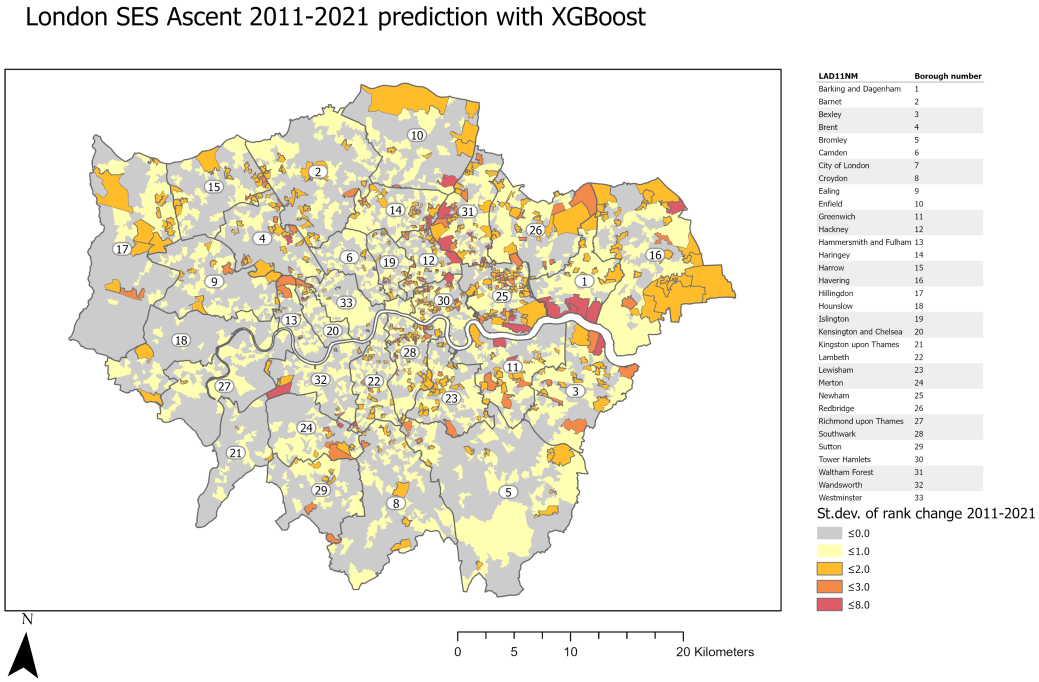



Figure 4.10: Future gentrification prediction with XGBoost.

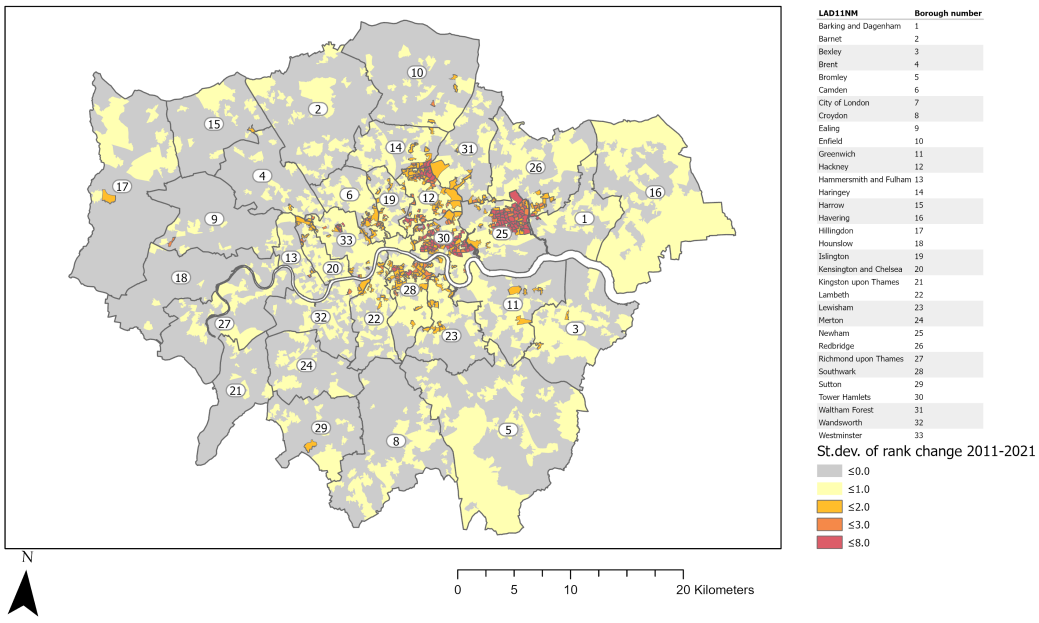



Figure 4.11: Future gentrification prediction with Random Forest.

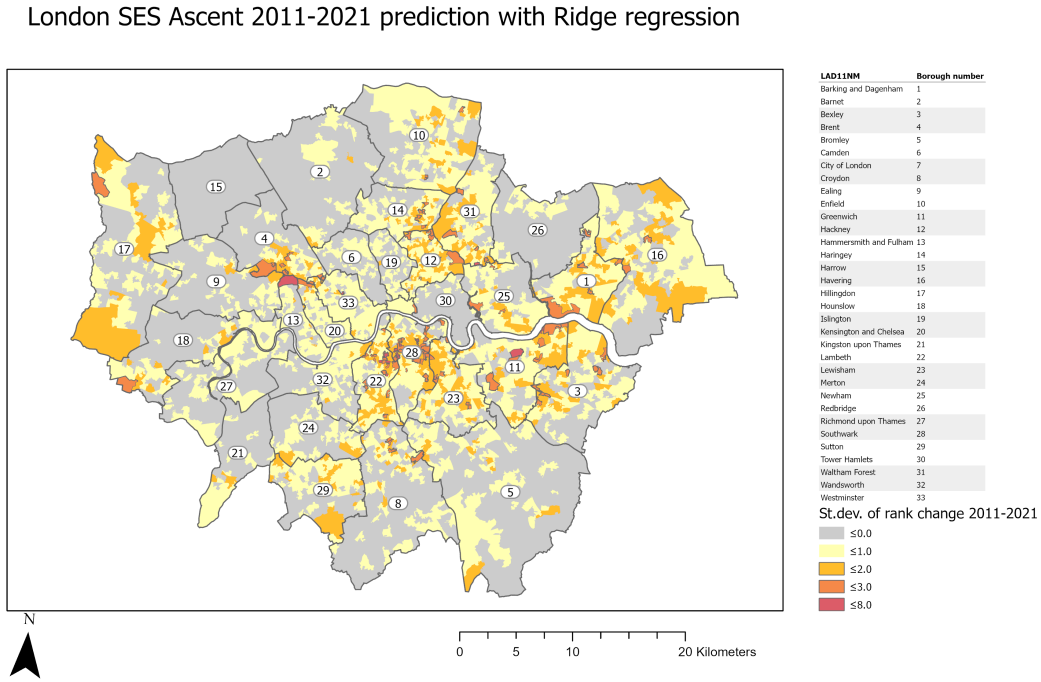



Figure 4.12: Future gentrification prediction with Ridge regression.

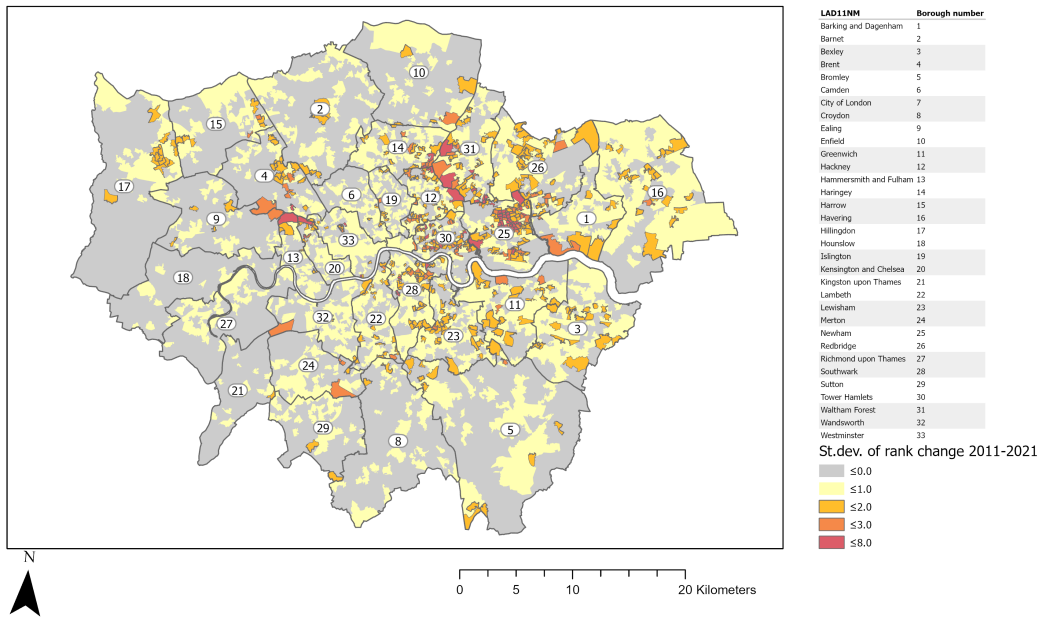



Figure 4.13: Future gentrification prediction with CatBoost.

The main research question of this thesis is: how robust is Machine Learning in the prediction and understanding of gentrification? We attempt to answer this question by focusing on three aspects, namely model performance, feature importance, and future prediction. The major findings from model performance are that most AI approaches seem to predict comparable or better than our linear regression baseline in terms of model fit statistics. The optimized Random Forest and Boosting algorithms XGBoost and CatBoost produce the most accurate results in terms of explained variance and error term. XGBoost appears to be the best candidate algorithm overall, since its runtime is relatively low while still having one of the highest evaluation scores. The main finding from the feature importance analysis is that the best performing models (XGBoost, CatBoost, Random Forest) have the same top 3 features, which means that the same features (House prices, Household Income, Males:49 or more hours) seem to have the highest average marginal impact overall. It seems likely that these three features are the most important when it comes to accurately predicting gentrification, and we also observe that the exact relation between feature value and model impact differs between similar models. Linear regression and other Machine Learning algorithms base their predictions on completely different features. Furthermore, SHAP appears to be a good approach to explaining geographic AI models. The most important finding from the future prediction analysis is that although Random Forest, XGBoost, CatBoost are similar in performance and feature importance, their future predictions of gentrification are very different when visualized. This raises questions for robustness of the approach and for practical applicability.

What do these findings mean? When it comes to algorithm performance, Reades et al. (2019) limit their machine learning focus to Random Forest regression, and compare it to linear regression baseline. The fact that boosting algorithms (XGBoost & CatBoost) perform just as good as Random Forest suggests that multiple Machine Learning approaches are possible for this task of gentrification prediction. Also experimenting with other algorithms such as Support Vector Regression (SVR), AdaBoost and K-nearest Neighbor (KNN) gives us a feel for how different algorithms perform at this task. The fact that AdaBoost has worse results than linear regression suggests that not all boosting algorithms are automatically better than our baseline and do not perform equally. If we introduce Big Data to a future approach, efficiency can become a factor that needs to be taken into account. XGBoost is the fastest algorithm with best performance (and runtime) overall, which makes it a suitable go-to candidate for bigger data tasks. The XGBoost algorithm has been reported to achieve state-of-the-art results on a wide range of machine learning tasks (Chen and Guestrin, 2016). Since it has also shown to perform well at predicting gentrification demonstrated in this thesis, we can therefore expect XGBoost to have potential for geographic Machine Learning tasks in general. The successful implementation of SHAP (Lundberg and Lee, 2017) allows us to interpret Machine Learning models, which are usually regarded as a so-called black box. Solving this current lack of interpretability in Machine learning (Doshi-Velez and Kim, 2017) is an important requirement for future geographic AI methodology, because SHAP enables geographers to gain understanding of why a predictive model predicts in the way that it does. Accurate interpretation of results is needed in order to effectively use this type of method for deeper theoretical understanding

and policy development. The gentrification visualizations show that every Machine Learning model that we have trained produces a unique future prediction; there are no easily distinguishable patterns shared between maps. This finding is surprising and unexpected, since Random Forest, XGBoost, and CatBoost are very similar in terms of performance metrics and with feature importance. The intuition was that this similarity between these three AI models would also result into a somewhat similar prediction of future gentrification, but this is not the case. This dissimilarity in prediction patterns is also not something observed or investigated by Reades et al. (2019), which means it is also not something that could be anticipated with literature research. This finding does however present a new challenge for predicting gentrification, because it raises questions for the validity and accuracy of our predictive models. Since our goal is to accurately forecast gentrification at a neighborhood level, and all of our models predict differently, which one should we then believe to be correct? This is not an issue pertaining to the robustness of Machine Learning per se, since predicting with a 'standard approach' such as linear regression presents us with the same situation. Instead, this difference in predictions suggests challenges with the methodology itself. Apart from the academic side, understanding and potentially minimizing this forecasting dissimilarity between models is also very important from a more practical perspective. If this Machine Learning approach as it is right now is used as a tool for policy-making (e.g. an implementation of Chapple and Zuk (2016)'s gentrification early warning system), there will be risks involved. It is not unthinkable that policy-makers will simply assume that the model forecast is highly accurate, resulting in ineffective or damaging policy decisions. Simply put, this gentrification forecasting approach still requires a significant amount of research and development before it can be deployed effectively in practical setting.

The core of our methodology is to operationalize gentrification in the form of Socio-Economic Status (SES) change at the neighborhood level as conceptualized by Owens (2012) and to then build a predictive model on two sets of data with a 10 year gap. The goal of this approach is that we end up with a machine learning model that is capable of accurately predicting SES changes. There can be two types of potential issues with this methodology: either the operationalization of gentrification is not effective, or the machine learning approach might require additional or different processing steps in order to produce useful results. When it comes to operationalization it is possible that Socio-Economic Status is not an ideal metric for visualizing future gentrification predictions, because it might be a too complex metric for the task. This challenge with operationalization is not unexpected, because gentrification is a complex multi-faceted phenomenon, encompassing many competing perspectives and explanations (Rigolon and Németh, 2019). Since most research in Human Geography is similarly complex, we can expect to encounter this type of operationalization challenge for other Geographic Data Science research as well. Generally, a clearly defined theoretical understanding of the phenomenon is necessary for robust interpretation of results. In the case of gentrification, the SES score used in our method is a composite value derived from Principal Component Analysis (PCA) on Household income, Median housing & sales, occupational share, and highest level of qualification. Choosing a more simple indicator and to model a certain aspect of gentrification might produce more reliable and uniform results. This selection of a suitable gentrification in-

dicator is not an easy task however, because data on many of the drivers and impacts of gentrification and displacement are not regularly gathered or are difficult to quantify (Chapple and Zuk, 2016).The overarching question here is whether we are predicting gentrification or something else that we incorrectly assume to represent gentrification. A less complex definition approach would allow for a simpler prediction task and more nuanced interpretation. Machine Learning tasks generally focus on predicting things that have a relatively straightforward "truth". Usually this pertains to tasks such as image classification or text extraction for which the target variable is quite easy to define: i.e., an image of a cat is an easy to define 'truth' and therefore requires not much further interpretation on a conceptual level. A complex phenomenon such as gentrification on the other hand is difficult to accurately define and operationalize; what our machine learning model ultimately is trying to predict is contingent on this operationalization. Standard machine learning goals focus on pattern recognition and prediction of relatively static entities such as text or images. Gentrification prediction on the other hand involves time and space. In other words, we try to predict the future and individual predictions are spatially related. The increased complexity of Machine Learning applied in geographic research appears to be much more of a challenge than traditional Machine Learning tasks, therefore possibly requiring the formulation of additional processing steps specifically for geographic machine learning. The other potential issue with our Machine Learning process is also related to this complexity of gentrification. The gentrification Machine Learning model is currently trained to predict a 10 year change in Socio-economic status for each individual neighborhood. The observed lack of common patterns between different algorithm predictions might indicate that this approach of predicting 10-year change is not the most robust way of forecasting gentrification. Quantitative research in Social Science is inherently about abstracting from reality, which means that the way in which we quantitatively interpret a geographic phenomenon is subject to design choices of the research method. The goal of quantitative research is to model and generalize the specific, which requires the researcher to make concessions in terms of data and theoretical conceptualization.

As is the case for most scientific studies, there is a series of limitations that can be identified for this thesis. The first one is that the performed case study on gentrification with ML does not give us an exact understanding of machine learning robustness in geographic research as a whole. This is because gentrification is only one topic of many topics, and so if we want to obtain a very refined overview of how machine learning performs in socio-economic research it will require comparison of more studies of different phenomena. Since the application of machine learning in non-traditional fields is still relatively new, not a lot of scientific literature on machine learning in geography available to make this literature comparison possible. Although in a strict sense this thesis only allows us to make assumptions on the robustness of machine learning when it comes to predicting gentrification, we can formulate more general expectations for Geographic Machine Learning. For example, the difference between global and local predictions as found in this thesis may translate to other geography problems as well. At the bare minimum, findings from this thesis can give us ideas on where to look for answers next.

The second limitation of this thesis' findings is for generalization at the spatial level; the study focuses only on gentrification prediction in the city of London. It is therefore unclear whether findings on machine learning ro-

bustness are also applicable for the prediction of gentrification in other UK cities, in other countries (e.g. Amsterdam, Berlin, Paris), or at a different spatial scale (regional or country-wide instead of city). We cannot assume that machine learning robustness of gentrification prediction will function the same way in different places, partly due to the inherent complexity of the concept of gentrification. A third limitation of our findings is that we only look at gentrification quantified in one specific way (via SES at neighborhood level), which means that findings on the method itself might not be generalizable. SES is perhaps not the most optimal approach to modelling gentrification due to it being a combination of several individual indicators. Another aspect of this method limitation is that the approach does not take into account gentrification effects of neighborhoods on other neighborhoods. Each neighborhood is treated as an individual prediction instead of it being modelled as part of an interconnected system. Our last identified limitation is that the data used in this study is not Big Data, which is high in volume, high in velocity (created in real-time), and exhaustive in scope (goal is to capture the entire population (Kitchin, 2014). Instead it consists of census data only. This study therefore does not provide any conclusive findings on how well machine learning performs at geographic research in a Big Data setting.

The four discussed limitations highlight a lot of possibilities for further research, such as experimenting with data for a different UK city, or using data from another country (Dutch, German, French, etc.). This will allow us to evaluate and compare prediction findings at a spatial level; what is the difference between global and local predictions when using Dutch data? Machine learning prediction with a different operationalization of gentrification will also be a valuable contribution, for example by using individual (or combined) gentrification indicators as defined by Chapple and Zuk (2016). Substituting or supplementing the current dataset with Big Data and comparing results will also generate new insights into modelling and subsequently accurately predicting gentrification. Glaeser et al. (2018) for example attempt to predict gentrification in New York by utilizing restaurant review data from the online platform Yelp. Another possibility is using geotagged Twitter data to model gentrification (Poorthuis et al., 2016). The inclusion of Big Data might also make it worthwhile to use a Neural Network for prediction. Ilic et al. (2019) for example use a technique called Deep Learning to map gentrification from visual characteristics of Google Street View data. This further research will enable us to better understand gentrification prediction and robustness of machine learning in geographic context.

# 5 CONCLUSION

This thesis has looked at how robust machine learning is when applied to geographic research. Although gentrification appears to be successfully modelled and predicted by machine learning at a surface level (performance metrics), when we explore future prediction and compare results in more depth we find that a lot of issues appear when it comes to the robustness and practical applicability of the executed method approach. The main contribution of this thesis is that it provides a concrete example of why researchers should be just as critical of machine learning results as with normal quantitative research. The findings from our study show that Machine learning is not some kind of magic bullet that compensates for this lack of domain knowledge and theoretical foundation. Anderson (2008) suggested that working with near-universal data sets and identifying patterns supplants the need for theory, but instead our results confirm the need for "intensified critical engagement of Data Science by geographers" (Singleton and Arribas-Bel, 2019, p. 2), since theoretical understanding is incredibly important for robust machine learning results in Social Science. A secondary contribution is the proposal and exploration of using Shapley Additive Explanations (SHAP) as a way of tackling the current lack of interpretation in machine learning models. Although the gentrification study from this thesis emphasizes the need for a critical approach of machine learning in geographic research, it certainly does not mean that machine learning has no place in geography at all. This is quite the opposite, as with the increasing availability of Big Data (Kitchin, 2014), Machine Learning and Data Science will be expected to play a much bigger role within Geography. What it does mean is that a lot of work is still to be done when it comes to robustness and practical implementation of something like an gentrification early-warning system such as proposed by Chapple and Zuk (2016). Graham and Shelton (2013) note that the "futures of geography and big data are still to be made". What this thesis shows is that in this future, domain knowledge remains at the centre. This means geographers will play the central role in further developing the field of Geographic Data Science.

# BIBLIOGRAPHY


Anderson, C. (2008). The end of theory: The data deluge makes the scientific method obsolete. *Wired magazine 16*(7), 16–07.

Anderson, E. (2013). *Streetwise: Race, class, and change in an urban community*. University of Chicago Press.

aporras (2016). What is the difference between bagging and boosting? [Online; accessed May 30, 2020].

Arribas-Bel, D. and J. Reades (2018). Geography and computers: Past, present, and future. *Geography Compass 12*(10), e12403.

Artstein, R. (2017). *Inter-annotator Agreement*, pp. 297–313. Dordrecht: Springer Netherlands.

Bardenet, R., M. Brendel, B. Kégl, and M. Sebag (2013). Collaborative hyperparameter tuning. In *International conference on machine learning*, pp. 199–207.

Barton, M. (2016). An exploration of the importance of the strategy used to identify gentrification. *Urban Studies 53*(1), 92–111.

Boehmke, B. and B. M. Greenwell (2019). *Hands-On Machine Learning with R*. CRC Press.

Brunsdon, C. (2016). Quantitative methods i: Reproducible research and quantitative geography. *Progress in Human Geography 40*(5), 687–696.

Caulfield, J. (1994). *City form and everyday life: Toronto's gentrification and critical social practice*. University of Toronto Press.

Chapple, K. and M. Zuk (2016). Forewarned: The use of neighborhood early warning systems for gentrification and displacement. *Cityscape 18*(3), 109–130.

Chen, T. and C. Guestrin (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794. ACM.

Cunningham, P. and S. J. Delany (2007). k-nearest neighbour classifiers. *Multiple Classifier Systems 34*(8), 1–17.

Datta, A., S. Sen, and Y. Zick (2016). Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *2016 IEEE symposium on security and privacy (SP)*, pp. 598–617. IEEE.

Doshi-Velez, F. and B. Kim (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

Easton, S., L. Lees, P. Hubbard, and N. Tate (2019). Measuring and mapping displacement: The problem of quantification in the battle against gentrification. *Urban Studies*, 0042098019851953.

Freund, Y., R. Schapire, and N. Abe (1999). A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence 14*(771-780), 1612.

Galster, G. and S. Peacock (1986). Urban gentrification: Evaluating alternative indicators. *Social Indicators Research 18*(3), 321–337.

Géron, A. (2017). *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.".

Glaeser, E. L., H. Kim, and M. Luca (2018). Nowcasting gentrification: using yelp data to quantify neighborhood change. In *AEA Papers and Proceedings*, Volume 108, pp. 77–82.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep learning*. MIT press.

Graczyk, M., T. Lasota, B. Trawiński, and K. Trawiński (2010). Comparison of bagging, boosting and stacking ensembles applied to real estate appraisal. In *Asian conference on intelligent information and database systems*, pp. 340–350. Springer.

Graham, M. and T. Shelton (2013). Geography and the future of big data, big data and the future of geography. *Dialogues in Human geography 3*(3), 255–261.

Gregory, D., R. Johnston, G. Pratt, M. Watts, and S. Whatmore (2011). *The dictionary of human geography*. John Wiley & Sons.

Guerrieri, V., D. Hartley, and E. Hurst (2013). Endogenous gentrification and housing price dynamics. *Journal of Public Economics 100*, 45–60.

Guyon, I. and A. Elisseeff (2003). An introduction to variable and feature selection. *Journal of machine learning research 3*(Mar), 1157–1182.

Harvey, D. (1985). *Consciousness and the urban experience: Studies in the history and theory of capitalist urbanization*, Volume 1. Johns Hopkins University Press.

Helbrecht, I. (2018). Gentrification and displacement. In *Gentrification and Resistance*, pp. 1–7. Springer.

Holm, A. and G. Schulz (2018). Gentrimap: A model for measuring gentrification and displacement. In *Gentrification and Resistance*, pp. 251–277. Springer.

Hu, Y., W. Li, D. Wright, O. Aydin, D. Wilson, O. Maher, and M. Raad (2019). Artificial intelligence approaches. *Geographic Information Science and Technology Body of Knowledge 2019*.

Ilic, L., M. Sawada, and A. Zarzelli (2019). Deep mapping gentrification in a large canadian city using deep learning and google street view. *PloS one 14*(3), e0212814.

James, G., D. Witten, T. Hastie, and R. Tibshirani (2013). *An introduction to statistical learning*, Volume 112. Springer.

Jordan, M. I. and T. M. Mitchell (2015). Machine learning: Trends, perspectives, and prospects. *Science 349*(6245), 255–260.

Kitchin, R. (2014). Big data, new epistemologies and paradigm shifts. *Big data & society 1*(1), 2053951714528481.

Lees, L. (2003). Super-gentrification: The case of brooklyn heights, new york city. *Urban studies 40*(12), 2487–2509.

Ley, D. (1994). Gentrification and the politics of the new middle class. *Environment and Planning D: Society and Space 12*(1), 53–74.

Lipovetsky, S. and M. Conklin (2001). Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry 17*(4), 319–330.

Lundberg, S. M., G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee (2019). Explainable ai for trees: From local explanations to global understanding. *arXiv preprint arXiv:1905.04610*.

Lundberg, S. M. and S.-I. Lee (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pp. 4765–4774.

Maurrasse, D. (2014). *Listening to Harlem: Gentrification, community, and business*. Routledge.

McKinney, W. (2011). pandas: a foundational python library for data analysis and statistics. *Python for High Performance and Scientific Computing 14*.

Molnar, C. (2019). *Interpretable machine learning: a guide for making Black Box Models interpretable*. Lulu.

Morde, Vishal (2019). Evolution of xgboost algorithm from decision trees. [Online; accessed May 30, 2020].

Oliphant, T. E. (2007). Python for scientific computing. *Computing in Science & Engineering 9*(3), 10–20.

Owens, A. (2012). Neighborhoods on the rise: A typology of neighborhoods experiencing socioeconomic ascent. *City & Community 11*(4), 345–369.

Park, B. and J. K. Bae (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.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research 12*(Oct), 2825–2830.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12*, 2825–2830.

Poorthuis, A., M. Zook, M. G. Taylor Shelton, and M. Stephens (2016). Using geotagged digital social data in geographic research. In N. Clifford, M. Cope, T. Gillespie, and S. French (Eds.), *Key methods in geography*, Chapter 16. Sage.

Reades, J., J. De Souza, and P. Hubbard (2019). Understanding urban gentrification through machine learning. *Urban Studies 56*(5), 922–942.

Ribeiro, M. T., S. Singh, and C. Guestrin (2016). Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*.

Rigolon, A. and J. Németh (2019). Toward a socioecological model of gentrification: How people, place, and policy shape neighborhood change. *Journal of Urban Affairs*, 1–23.

Rose, D. (1996). Economic restructuring and the diversification of gentrification in the 1980s: a view from a marginal metropolis. *City lives and city forms: Critical research and Canadian urbanism*, 131–172.

Sayad, Saed (2012). Support vector machine - regression (svr). [Online; accessed May 30, 2020].

Scikit-learn (2011). Supervised learning overview. [Online; accessed May 27, 2020].

Shrikumar, A., P. Greenside, A. Shcherbina, and A. Kundaje (2016). Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*.

Singleton, A. and D. Arribas-Bel (2019). Geographic data science. *Geographical Analysis*.

Smith, N. (2005). *The new urban frontier: Gentrification and the revanchist city*. Routledge.

Smith, N. and M. Sorkin (1992). *New city, new frontier: the Lower East Side as wild wild west*. Noonday Press.

Štrumbelj, E. and I. Kononenko (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems 41*(3), 647–665.

Walks, R. A. and R. Maaranen (2008). Gentrification, social mix, and social polarization: Testing the linkages in large canadian cities. *Urban Geography 29*(4), 293–326.

Wang, X., J. Wen, Y. Zhang, and Y. Wang (2014). Real estate price forecasting based on svm optimized by pso. *Optik-International Journal for Light and Electron Optics 125*(3), 1439–1443.

# A PARAMETER SETTINGS

Table A.1: Settings for optimized XGBoost

| Hyperparameter | Value |
|---|---|
| random_state | 0 |
| max_depth | 3 |
| learning_rate | 0.25 |
| gamma | 0.1 |
| min_child_weight | 4 |
| n_estimators | 200 |
| tree_method | 'hist' |

Table A.2: Settings for optimized Random Forest

| Hyperparameter | Value |
|---|---|
| bootstrap | False |
| criterion | 'mse' |
| max_depth | None |
| max_features | 0.85 |
| max_leaf_nodes | None |
| min_impuritiy_decrease | 0.0 |
| min_impurity_split | None |
| min_samples_leaf | 2 |
| min_samples_split | 2 |
| random_state | 42 |
| n_estimators | 1400 |
| n_jobs | -1 |
| oob_score | False |
| warm_start | False |

Table A.3: Settings for optimized K Nearest Neighor

| Hyperparameter | Value |
|---|---|
| algoritm | 'ball_tree' |
| leaf_size | 4 |
| metric | 'minkowski' |
| metric_params | None |
| n_jobs | None |
| n_neighbors | 15 |
| p | 2 |
| weights | 'distance' |

Table A.4: Settings for optimized Support Vector Regression

| **Hyperparameter** | **Value** |
|---|---|
| C | 3 |
| cache_size | 200 |
| degree | 3 |
| epsilon | 0.1 |
| gamma | 'auto' |
| kernel | 'rbf' |
| max_iter | -1 |
| shrinking | True |
| tol | 0.001 |
| verbose | False |

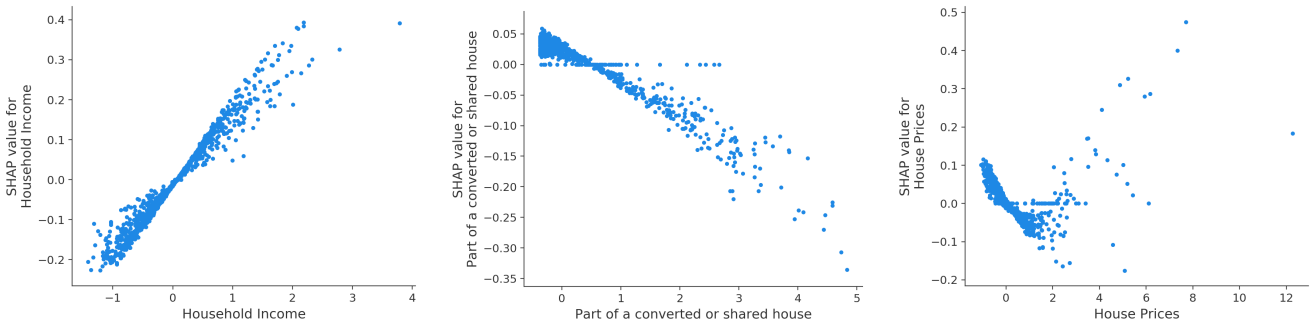
Table A.5: Settings for optimized AdaBoost Regression

| **Hyperparameter** | **Value** |
|---|---|
| base_estimator | None |
| learning_rate | 0.2 |
| loss | 'exponential' |
| n_estimators | 50 |
| random_state | 0 |

# B

# SHAP RESULTS

## B.1 SVR




(a) Rank 1 - Household Income

(b) Rank 2 - Part of a converted or shared house

(c) Rank 3 - House Prices

Figure B.1: Dependence Plots: SVR regression

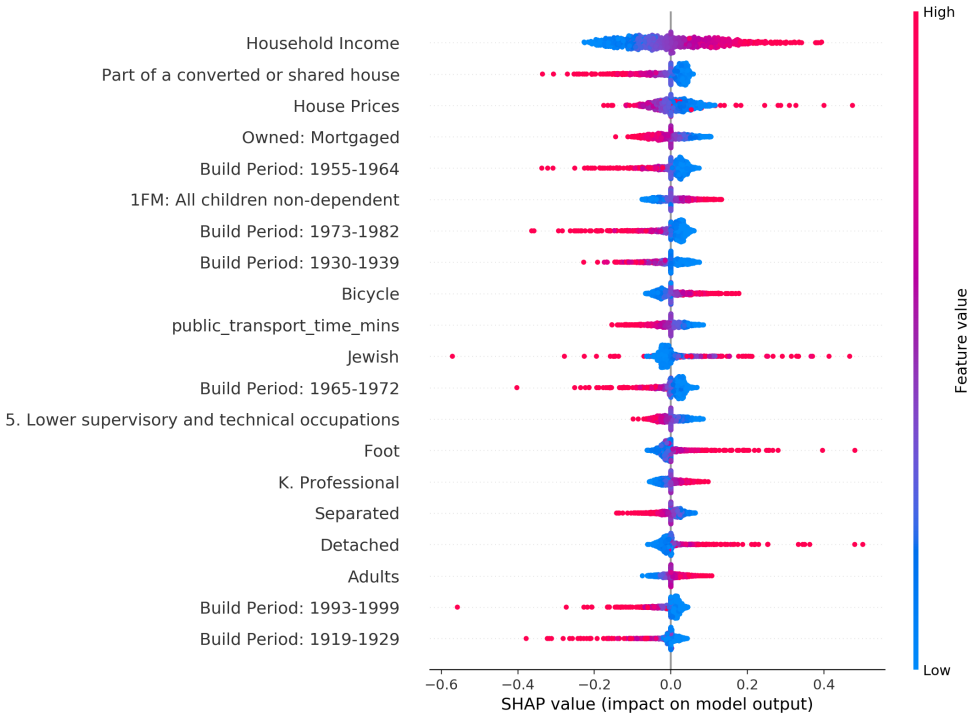
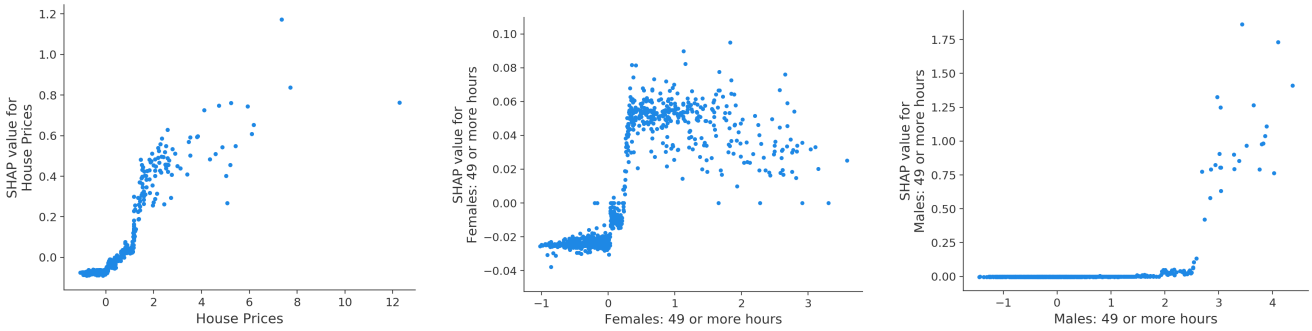



Figure B.2: SHAP summary plot for Support Vector regression that takes into account intensity and direction of feature impact.




(a) Rank 1 - House Prices (b) Rank 2 - Females:49 or more hours (c) Rank 3 - Males:49 or more hours

Figure B.3: Dependence Plots: AdaBoost regression

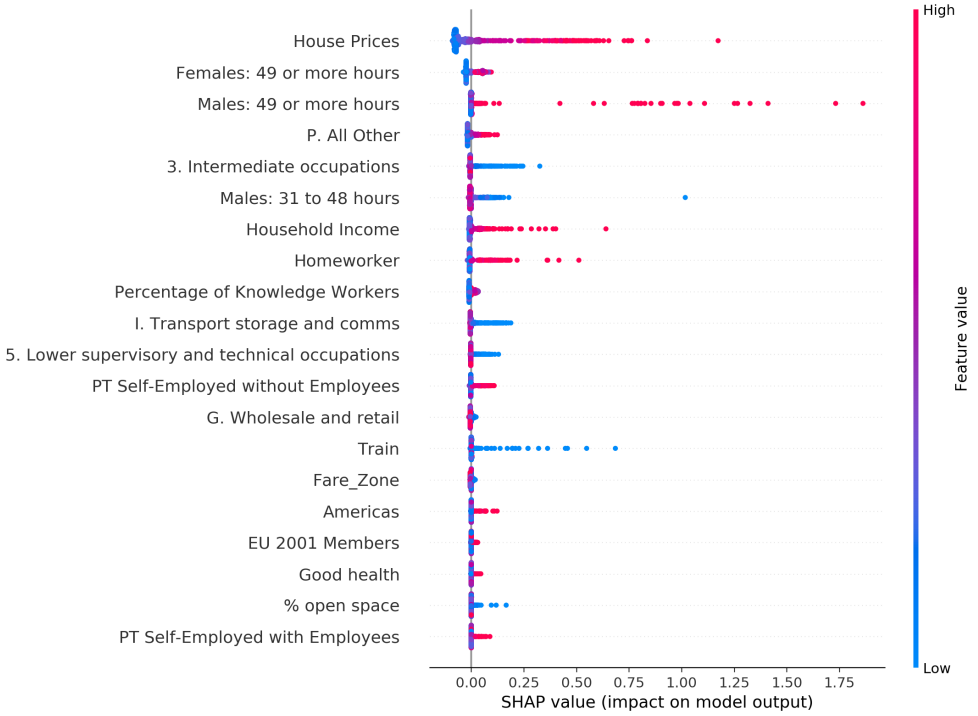
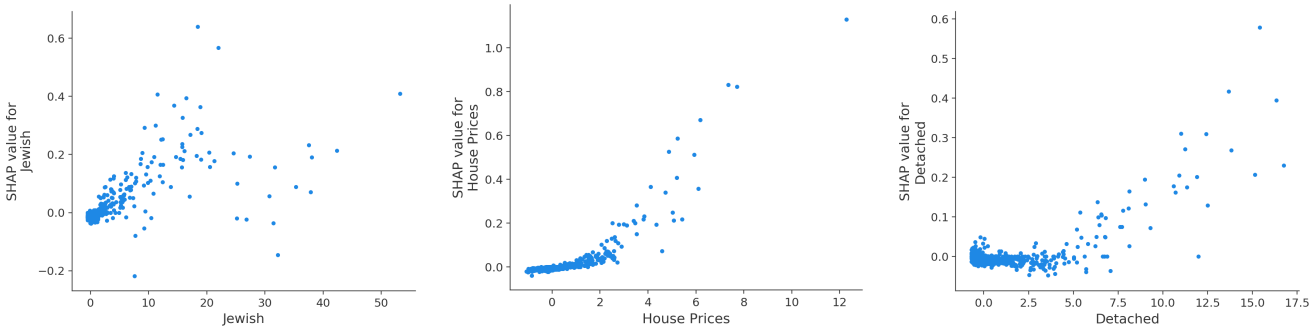



Figure B.4: SHAP summary plot for AdaBoost regression that takes into account intensity and direction of feature impact.




(a) Rank 1 -Ethnicity: Jewish (b) Rank 2 - House Prices (c) Rank 3 - Housing style: Detached

Figure B.5: Dependence Plots: KNN regression

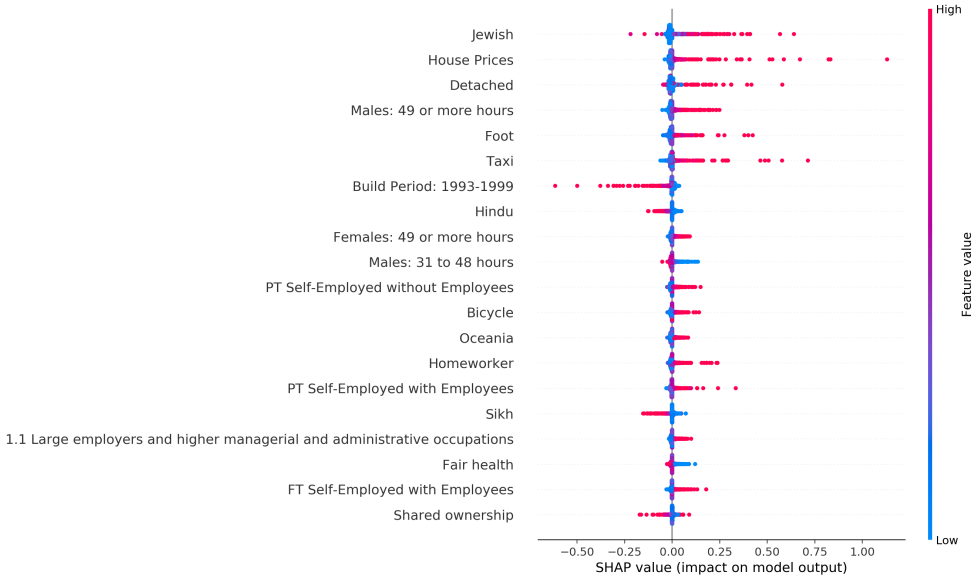
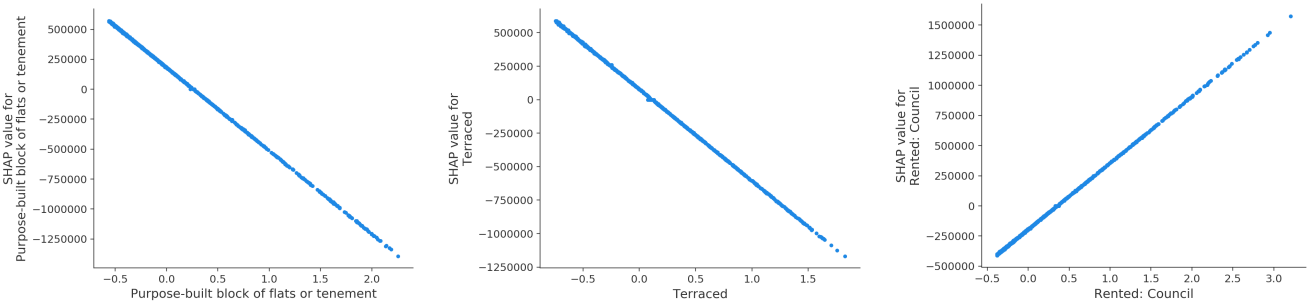



Figure B.6: SHAP summary plot for KNN regression that takes into account intensity and direction of feature impact.




(a) Rank 1 - Purppose-build blocks (b) Rank 2 - Terraced (c) Rank 3 - Rented: Council

Figure B.7: Dependence Plots: Linear regression

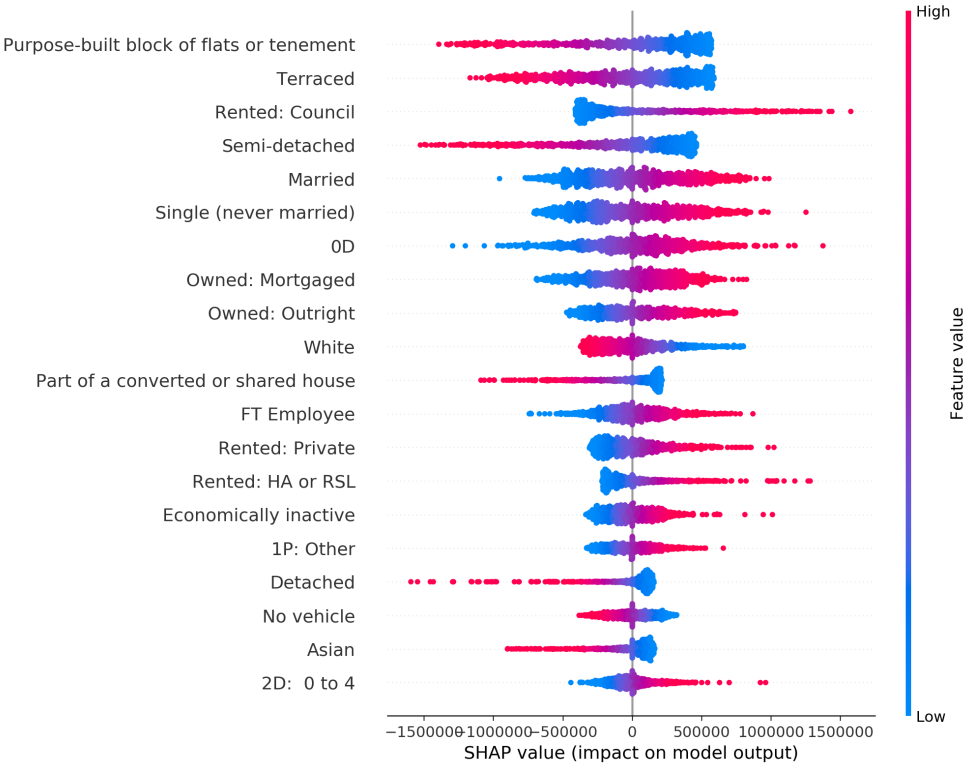
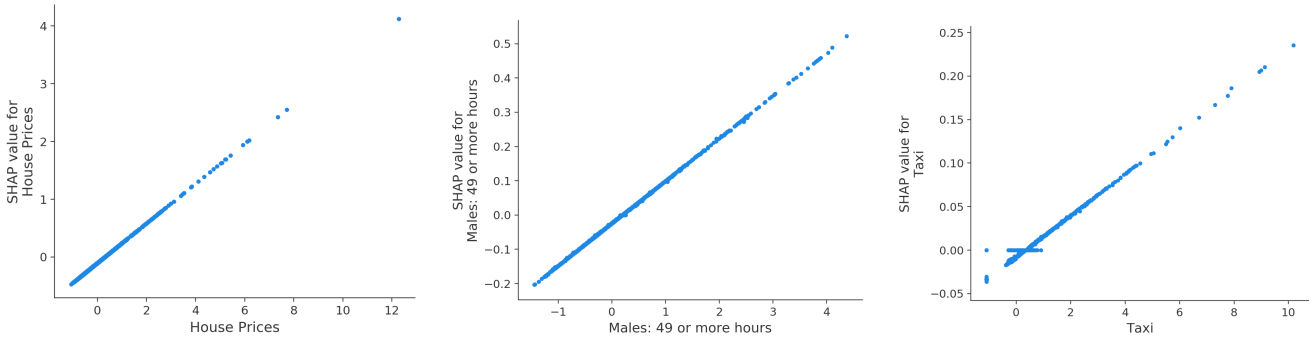



Figure B.8: SHAP summary plot for Linear regression that takes into account intensity and direction of feature impact.




(a) Rank 1 - House Prices (b) Rank 2 - Males:49 or more hours (c) Rank 3 - Taxi

Figure B.9: Dependence Plots: LASSO regression

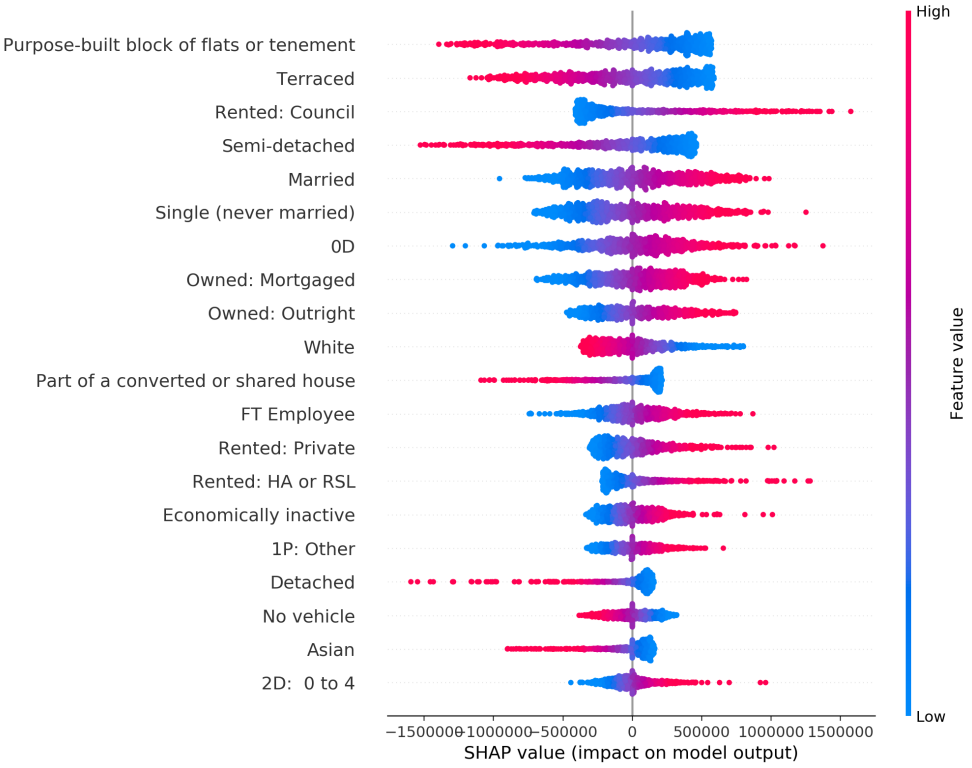



Figure B.10: SHAP summary plot for LASSO regression that takes into account intensity and direction of feature impact.

# C

## SES RANK CHANGE HISTOGRAMS

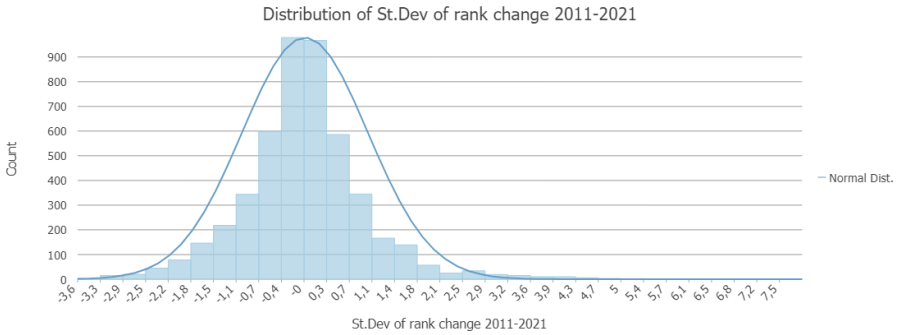



Figure C.1: Histogram of SES Ascent prediction scores with CatBoost.

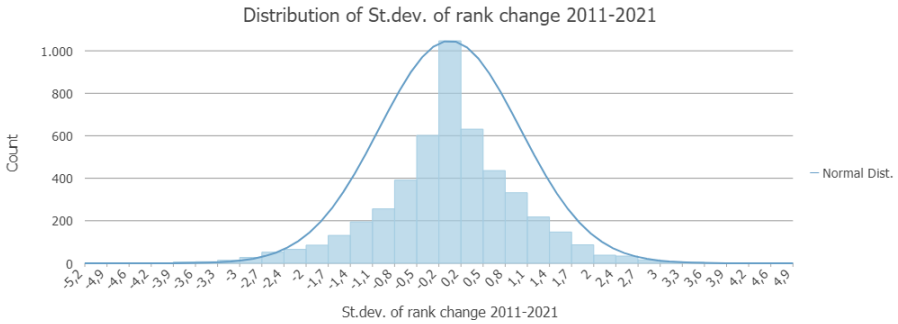



Figure C.2: Histogram of SES Ascent prediction scores with Ridge regression.

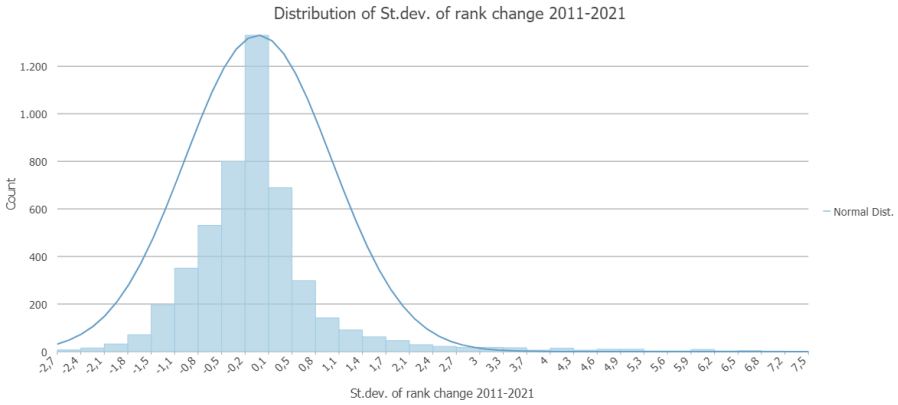



Figure C.3: Histogram of SES Ascent prediction scores with Random Forest.