Meyberg, Camilo; Rendtel, Ulrich; Leerhoff, Holger

Flat rent price prediction in Berlin with web scraping

AStA Wirtschafts- und Sozialstatistisches Archiv

Provided in Cooperation with:
Springer Nature

Suggested Citation: Meyberg, Camilo; Rendtel, Ulrich; Leerhoff, Holger (2024) : Flat rent price prediction in Berlin with web scraping, AStA Wirtschafts- und Sozialstatistisches Archiv, ISSN 1863-8163, Springer, Berlin, Heidelberg, Vol. 18, Iss. 2, pp. 245-278, https://doi.org/10.1007/s11943-024-00340-6

AStA Wirtschafts- und Sozialstatistisches Archiv (2024) 18:245–278
https://doi.org/10.1007/s11943-024-00340-6

ORIGINALVERÖFFENTLICHUNG

Flat rent price prediction in Berlin with web scraping

Camilo Meyberg · Ulrich Rendtel · Holger Leerhoff

Received: 6 October 2023 / Accepted: 28 May 2024 / Published online: 24 June 2024

Abstract Internet data pose a challenge to the traditional system of official statistics, which relies on more conventional sources such as surveys and registers, not readily adaptable to rapid changes. Expanding this system to include internet data is currently at an experimental stage, exploring these sources' potentials and benefits. This paper describes a project conducted within the ESSnet *Trusted Smart Statistics – Web Intelligence Network* framework. It investigates the use of online apartment listings to analyze the rental market. We used web scraping to extract information from two online real estate portals for flats in the city of Berlin. Using this data, we developed a model to predict rental prices per square meter based on the accommodation's features and location within the city. We detected offers which appear in both portals by means of statistical matching and removed duplicate offers. Missing values were treated by multiple imputation. The prediction model is a semi-parametric approach where the postal districts are used to describe the location effect. Comparisons with microcensus results and the local rent index reveal significant differences between the market of online flat offers and the stock of existing flat contracts. Interested readers will find the commented programming code in the internet supplement.

Keywords Web scraping · Flat offers · Official statistics · Rent indices · Statistical matching · Multiple imputation · Semi-parametric regression

✉ Camilo Meyberg · Ulrich Rendtel
Freie Universität Berlin, Berlin, Germany
E-Mail: camilo.meyberg@gmail.com

Ulrich Rendtel
E-Mail: ulrich.rendtel@fu-berlin.de

Holger Leerhoff
Amt für Statistik Berlin-Brandenburg, Berlin, Germany
E-Mail: holger.leerhoff@Statistik-BBB.de

JEL classification R21 · R31 · L86

1 Introduction

The production of modern official statistics is based on a system of scientific methods, regulations, codes, practices, ethical principles, and institutional settings that was developed in a pre-digital world, where surveys and administrative registers were the most important data sources for decades. Data in those times were a luxury and its acquisition was very costly. This means that most of the system was designed with this kind of data source in mind (Ricciato et al. 2020).

Nowadays, with the digitalization of society, there are new digital data sources available for official statistics that offer improved timeliness, accuracy, finer temporal and spatial resolution, more detail, increased relevance, possibly lower production costs, optimal statistics production and also competition with private players that provide statistics from a different outlook using this kind of data. This implies a challenge for official statistics since it has to adapt to these "new digital data".

The ESSnet project *Trusted Smart Statistics – Web Intelligence Network* is an evolution of activities from the former ESSnet Big Data I and II projects (2016–2020), aiming to generate multi-purpose statistics enhancing the integration of web data into official statistics through a comprehensive, coordinated approach rather than isolated efforts by individual National Statistical Institutes. Key objectives include increasing knowledge and competencies in web intelligence, advancing developments in specific domains for statistical production integration, identifying new web data sources for potential integration, and developing a robust business architecture, along with methodological and quality frameworks equipped with quantitative quality indicators for producing statistics with web data. Work Package 3 of the project focuses on investigating the potential of novel web data sources and the generation of experimental statistics derived from these sources (European Commission 2024). This article presents findings from a subsidiary research effort within the framework of Work Package 3 of the project.

While the importance of the real estate market on macro-economic indicators is well researched (Agnello and Schuknecht 2011; Catte et al. 2004; André and Girouard 2009; Leamer 2007; Bergenstrahle 2016), the rental market has received less attention. Despite this fact, its importance and relevance in the macro- and microeconomic context has been pointed out by some researchers (Arregui et al. 2009; Arce and López-Salido 2011; Caldera and Andrews 2011; Kofner 2014; Bergenstrahle 2016; Czerniak and Rubaszek 2018; Rubaszek and Rubio 2020) which is highly related to the objectives of official statistics where the rent prices are covered in the consumer price index analysis. Recently Steorts et al. (2020) have used online flat prices to analyze regional price differentials at the level of neighborhoods. In this context, data from official statistics could not deliver the regional information at this low regional level. In Germany the only source from official statistics with detailed information on flats is the Microcensus subsample on accommodations which is repeated every four years, see Frink and Rendtel (2019). Schmandt (2021) used this information on housing together with information on income for an analysis

of the affordability of flat prices. As the share of the flat rate on the disposable household income can amount up to 50 percent (Schmandt 2021) flat rates are of general public interest. In Germany there are official local rent indices, the so-called "Mietspiegel", which measure regional flat rates at a city level, see Sebastian and Memis (2021). Based on survey selection they are updated every two years. The quality of the Berlin rent index has been reviewed by Frink and Rendtel (2019).

However, the rent indices mainly address existing rent contracts and exclude newly built accommodations by law. Thus, there is no up-to-date data source that monitors the market for new contracts. Here online advertisements of flats offer a chance to fill this gap.

In this study, we will investigate this source for the Berlin rental market. We will scrape online offers from two online portals. Our aim is to predict the price per square meter given the relevant characteristics of the accommodation, including its location in the town. We will compare these flat rates with the data from the actual Berlin rent index.

In this article, we document the necessary steps to achieve this goal: The scraping of online offers from the two portals (Sect. 2), the preprocessing of the rent data (Sect. 3), the identification of flat offers that appear on both platforms and the necessity to de-duplicate these offers (Sect. 4), the preparation of the merged data set (Sect. 5), the treatment of missing values by multiple imputation (Sect. 6) and, finally, the presentation of the semi-parametric regression model for the flat rate per square meter (Sect. 7). In the conclusions (Sect. 8) we summarize and point to some remaining problems, like the representativeness of data from the portals, and possible measurement problems that may occur by confusing the different components of flat costs.

Readers who are interested in the details of our work and the programming code will find a link to a Jupyter Notebook of our analyses in the appendix.

2 Scraping flats offers

Two of the largest real estate portals in Germany were selected to be scraped: *Immowelt* and *Immonet*. A web scraper for each portal was developed using Scrapy which is a Python application framework for writing web spiders that crawl websites and extract structured data from them (Scrapy developers 2022a). For this task, configuring and defining spiders that navigate through the portals is necessary. Spiders are programs, called classes, that define how a certain page or group of pages will be scraped, how to follow the links (crawl), and how to extract and parse structured data from their pages. In other words, the whole process is set up by them.

The general logic of the procedure is as follows: Suppose a data analyst is in front of their computer and wants to explore and extract offers from an online real estate portal. The first task is to start their search on the portal by fixing all the criteria for the search in the search menu. Then they are redirected to a page that shows a list of online offers according to their search query. This URL is the first page the scraper will visit. Then they start to scroll the web page and create a table where they store the ID of the offer, the name, some general information, and the URL which

leads to the description of each specific offer, the so-called exposé. After scrolling down the whole page, the analyst notices that there are still more offers available on the following pages. Therefore, they proceed to the next page and repeat the same procedure until they have read the last offer on the last page. This is the job of the first group of spiders, which is in charge of extracting and exporting all the available offers to a list in CSV format. Additionally, a check for repeated offers that should be excluded from the list is performed.

At this stage, the analyst has a list with each offer and the corresponding URLs to the corresponding exposés. Now, their next step is to extract the specific information they need from each exposé. Here, they explore the description of the offer and collect the available relevant information. Then they select and copy this information, which is then added to a new table that will contain the specific information of all offers. They clean and format the obtained data according to their needs. Some basic examples and a step-by-step guide for beginners can be found in the official tutorial of the Scrapy package (Scrapy developers 2022b).

Since both portals are well structured, it was relatively easy to use selectors that locate and extract the relevant information. For this purpose, the inspector view of the web browser Google Chrome allows one to check and inspect the chunks of code and their hierarchy. Immowelt offers the advantage by granting access to an API that retrieves the whole information of the exposé as a JSON file through the source code of the page. This was not the case for Immonet, where each attribute had to be localized separately. This also necessitated a preliminary removal of some symbols embedded in the HTML code.

The data were scraped between May and October of 2022. While the information from Immonet was scraped on a weekly basis, the offers from Immowelt were extracted on a daily basis. In our analysis, we did not consider the time stamp of the offer. However, if available, this kind of information can be helpful to identify duplicates of offers on different portals.

3 Preprocessing of the data

The scraped data must be prepared to be used for later analysis. Each dataset representing a real estate portal was preprocessed separately. At this stage, the biggest challenge was to establish the comparability of the data across the portals.

The two portals present their offers in their own specific manner. This implies that the structure, data presentation, and method of data input can vary between different portals. While one portal has standardized the information rigidly, the other allows more flexibility at the moment a new offer is published.

While Immonet is more strict with the type and values that variables can take, Immowelt is the opposite case. The latter portal did not restrict the phrasing of some string variables. As a consequence, different string combinations could have the same meaning. For example, the deposit for the flat rate was reported in some cases as three times the monthly rent price, while in other cases, it was reported in Euro.

A list of the scraped variables can be seen in Table 12. Additionally, the corresponding step-by-step preprocessing of each portal with an overall diagnostic analysis can be found in the internet supplement (Jupyter Notebook, Appendix A and B). Here, the empirical distributions of the two portals are quite similar to each other.

Even if there was a relatively high number of offers, the number of offers that were useful for the final model was as low as 43% (Immowelt) or 66% (Immonet) of the total available cases.

Duplicates were also considered at this stage. In this context, there are two types of duplicates. The first kind of duplicates are flats that are published on the portal repeatedly with the same information under unique identification numbers. The second type are offers that are published with minor changes and updates of their attributes while the rest remains constant. However, for some reason, they receive a new identification number. The first type of duplicate is easy to filter. Hence, these were removed, and only the latest duplicate offer was retained. However, the second type is not easy to identify. Using record linkage (See Sect. 4 below) in one of the datasets, it was observed that the proportion of these cases is not higher than 5% of the number of offers. Keeping these offers in the dataset and regarding them as new offers should not have a major impact on the estimation of statistical models and their inference.

The different portals may attract different flat offers. As we are interested in an analysis of the flat rate, we compare the average of the flat rates across the two portals. Table 1 states a substantial difference in the mean flat rate of the portals. Here the Immonet flats are on average 3.13 Euro more expensive than the flats on the Immowelt portal. We conclude that one should scrape from both portals in order to avoid selective effects with respect to the flat rate.

As we collect offers from the two portals, it may happen that one flat is offered on both platforms. If we want to analyze the data from both portals, we have to merge the separate datasets and determine the duplicates that occur in both portals. At this step, it is important to identify and de-duplicate the offers from the joint database. The identification of duplicates is relevant for better estimations of the size of the online market as well as to reduce a potential bias that might appear in the statistical analysis of rent prices. Here we conjecture that offers above average need more advertising. Because of the high methodological impact of de-duplication, which is documented in the subsequent sections, we already present the empirical result of the de-duplication in Table 2. Approximately one-third of the 2950 analyzed offers

Table 1 Comparison of flat rates across portals

Real Estate Portal	Average Rent Price per m^2	Standard Deviation	Number of Observations
Immonet	15.32	5.68	1528
Immowelt	12.19	5.16	2543

Table 2 Empirical De-duplication results

Type of Observation	Average Rent Price per m^2	Standard Deviation	Number of Observations
Duplicates	14.52	5.40	1084
Non-Duplicates	13.14	5.61	1866

appear in both portals, and the duplicate offers are significantly more expensive, averaging 14.52 Euros per m^2 compared to 13.14 Euros per m^2 for offers on only one platform.

Therefore deduplication is a necessary step to achieve more reliable results for the online flat market. In the subsequent sections, we use record linkage techniques to identify identical flats in different portals.

4 Record linkage

The term "record linkage" refers to the procedure of linking records of two or more different sources that are believed to correspond to each other. It can also be used to identify duplicates. The idea behind this concept is to use attributes that characterize an entity. These attributes are then used to link units in different records. The linkage procedure can be represented as a workflow (Christen 2012) that consists of the following phases: Preprocessing, Indexing, Comparing, Classification, and Evaluation. The Record Linkage Toolkit is a Python package that has been developed with the essential tools for record linkage and de-duplication and also allows for customized procedures (de Bruin 2022c).

4.1 Preprocessing

Preprocessing encompasses all the data cleaning and standardization required to increase the accuracy of record linkage. Although the package offers separate tools for this task, this step was already done as described in Sect. 3.

4.2 Indexing

Indexing is used to generate the record pairs for subsequent comparisons. These pairs can also be referred to as candidate matches. The naive approach to indexing is to generate all possible pair combinations for comparison (full indexing). However, this is not computationally efficient. For this reason, there are some indexing algorithms available that can efficiently reduce the number of generated candidate matches. In our case, block indexing was used.

Block indexing restricts the candidate matches to pairs that agree on a set of one or more variables called blocks. The postal code of each offer was used as a block. More detailed information for indexing background can be found in Christen (2008) and Christen (2012). The use of the postal code for blocking was highly efficient. The reduction was 99% of the original candidate matches.

4.3 Comparing

It is important to select a set of informative, discriminating, independent, stable, and plausible features for a good classification model. These features have a huge impact on the performance and reliability of the identification of duplicates. The main idea of comparison is to set and calculate a similarity measure along the

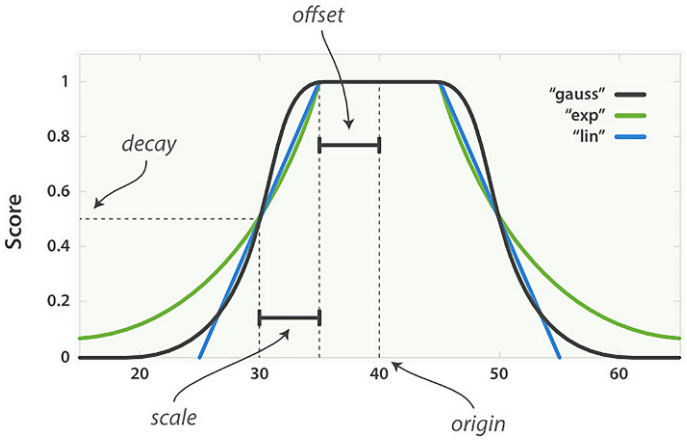

Fig. 1 Example of Decay Functions (Elasticsearch B.V. 2022a)

candidate matches with respect to the set of selected variables. There are some developed techniques according to the type of variable and the kind of comparison to perform.

The default options for dissimilarity measures are the Levenshtein distance for string variables and the linear decay function for numeric variables. The Levenshtein distance is also widely referred to as the "edit" distance which denotes the minimum number of single-character changes (insertions, deletions, or substitution) that a string "A" requires in order to be transformed into a string "B". This distance can be scaled so that the range lies between 0 and 1 by dividing the obtained distance by the maximum possible distance between the two compared strings.

Decay functions are used by the open and free platform ElasticSearch (Elasticsearch B.V. 2022b). Decay functions compute similarities of numeric fields according to their numeric differences. There are tolerated differences, called offset. Large differences are scaled by a decay rate, see Fig. 1. This methodology is implemented in the Python Record Linkage Toolkit Package. The default decay function is the linear one (de Bruin 2022b).

Thus each variable can contribute a maximum value of 1 to the joint similarity measure. Thus, the similarity index ranges between 0 and the number of comparison variables, which is 7 in our application.

In the case of the presence of missing data, most classifiers cannot handle comparison vectors with missing values. To prevent these problems the similarity index is set to its minimum value in case of missing values which reduces the risk of identifying false positives at the expense of increasing the probability of identifying false negatives (Ong et al. 2014). This approach for handling missing values is widely used in record linkage applications (de Bruin 2022c).

We selected the following seven variables for the similarity measure: the postal code, the address, the price of the rent, the utility costs, the heating costs, the living area, and the number of rooms. The postal code was used as a block previously. However, it was included as a comparison feature due to the fact that it is not

plausible that two flats in different neighborhoods are identified as equal. For the string and numerical variables the default methods and settings of the package were used. However, for the number of rooms and the postal code, we required an exact match.

4.4 Classification

The candidates are classified as matches and not matches by using machine learning methods. Machine learning can be supervised or unsupervised. The first approach requires a training dataset which is not required for the unsupervised approach. In order to have a proper training set, this must consist of a list of comparison pairs with a known correct labeling. In this particular case there is no such dataset available. Hence, the focus will lie on unsupervised methods.

The Record Linkage Toolkit offers two methods for unsupervised learning: the k-means classifier and the Expectation Maximisation (EM) classifier. The k-means classifier splits the candidates into match and non-match clusters, where the candidates are located in the cluster with the nearest mean. This method minimizes the within-cluster variance. The EM classifier uses a model for the latent unobserved class membership. For given values of the class membership the expected values of the likelihood is computed (E-Step). In the next step, the M-Step, the expected likelihood is maximized. This process yields new values for the class membership. This procedure is repeated until convergence is reached, see, for example, Collins (2012). We use here a classification of pairs into only two clusters: Either they are a match or they are not. There is also the alternative of assigning each couple to one of three different clusters (Elfeky et al. 2002): Match, Non-match or Possibly Match. For this reason we also refrain from using Hierarchical Clustering, for example Complete Linkage, where the number of clusters is to be estimated (Mamun et al. 2016).

It is still a good idea to evaluate the results from the unsupervised learning on a subset of observations where the correct classification can be labeled with a high degree of plausibility. However, such a small "training" set should be distinguished from a real training set with correct labels for all observations. Therefore we refrained from using this evaluation "training" set in a supervised learning algorithm for the entire sample.

This evaluation "training" dataset was constructed manually as follows:

1. Using the overall similarity scores from Sect. 4.3 the set of comparison pairs is split into four disjoint groups:
   - Comparison pairs with an overall score of 7.
   - Comparison pairs with an overall score between 6 and 7.
   - Comparison pairs with an overall score between 4 and 6.
   - Comparison pairs with an overall score below
2. A random sub-sample of 500 comparison pairs in each group was taken and then merged into an stratified sample of 2000 pairs. This represents 0.5% of the total of the comparison pairs. Labeling a bigger sample manually in a short time was not feasible.

Table 3 Example of a positive labelled comparison pair

ID	Street	Postal Code	Rent Price	Utility Costs	Heating Costs	Living Area	Rooms Number
48512451	NaN	12205	€1700	€226	NaN	140 m^2	4
27HL557	NaN	12205	€1700	€226	NaN	140 m^2	4

3. A so-called annotation file is created. This file contains the information of each comparison pair together with an additional field where the label of each pair is to be stored manually.
4. In our analysis the labeling takes place in a browser-based user interface for the manual classification of the comparison pairs called the RecordLinkage ANNOTATOR. A hosted version of this interface can be found on Github (de Bruin 2022a).
5. Afterwards, the interface shows each comparison pair with the respective information. The user clicks either on the button "match" or "distinct" to label the pairs.
6. The labeling was done under the following criteria:
   - If all the observed attributes coincide exactly, the comparison pair is labeled as a match. Note that missing attributes in both offers are ignored here.
   - If there are some missing values in one offer, but there is full agreement in the available attributes with the other offer, the comparison pair is labeled as a match.
   - If the address and/or the ZIP code of the offers do not coincide, the pair is labeled as distinct.
   - If there are negligible differences in any of the attributes of the offers, the comparison pair is labeled as a match.
   - For the rest of the cases, the comparison pairs were labeled as distinct.
7. Labelling continues until the interface confirms that all comparison pairs have been labelled Then, the labels are imported to Python and appended to the trained dataset.

An example of one positive labeled comparison pair can be seen in Table 3. Despite the fact that there is some missing information, this couple of offers is recognized as a match due to their coincidences in the values of the observed variables.

For the evaluation of the classification, we used the confusion matrix of the classification of the training data set. The performance of the models was evaluated using the standard indicators: Accuracy, recall, precision, sensitivity, specificity, and the F-score (Goutte and Gaussier 2005; Botchkarev 2019). It is desirable to achieve high values for all of them.

4.5 Record-Linkage results

The results of the confusion matrix given by the k-means and the EM classifiers can be seen in Table 4. The values of all cells sum up to the 2.000 comparison pairs from the training sample.

Table 4 Confusion matrix of the unsupervised methods

K-Means Classifier		
	Predicted Positives	Predicted Negatives
True Positives	982 (TP)	2 (FN)
True Negatives	413 (FP)	603 (TN)
Expectation Maximization Classifier		
	Predicted Positives	Predicted Negatives
True Positives	982 (TP)	2 (FN)
True Negatives	454 (FP)	562 (TN)

The estimated confusion matrices show that both methods are good at identifying positive matches at the expense of labeling negative matches as positive. Thus the general automatic classifiers, k-means as well as the EM-classifier, are less restrictive in recognizing differences compared to the manual classification.

For this reason, we display here three examples of false positive classification. Table 5 displays an example where the number of streets varies while all other variables are equal. Such cases are realistic if there are several new similar buildings in the same street. However, the Levenshtein distance counts only 1 change to make the 15 characters of the variable address fit. As all the other variables fit exactly, the general classifier recognizes these offers as possibly the same offer.

In Example 2, we have two deviations in numeric variables, see Table 6. While the difference in rent price might be regarded as of minor importance with respect to the offset of the decay function, the difference in the living area is substantial from our customized rules. Here a substantial difference in one variable is enough to reject the classification as a pair. However, the general classifiers also use the information from the other variables. In this case, 6 out of 7 distance measures are 0.

Finally, Table 7 displays a situation where the address number is different and also three numerical variables vary, but with minor changes. Also here the offset of the decay function will ignore the numerical differences and the length of the address with length 17 is large compared to the 2 necessary changes that the two offers are away from each other.

Table 5 False Positives: Example 1

ID	Street	Postal Code	Rent Price	Utility Costs	Heating Costs	Living Area	Rooms Number
47855741	Wustrowerstr. 11	13051	€829	€85	€85	71.2 m^2	3
26JQF5Q	Wustrowerstr. 17	13051	€829	€85	€85	71.2 m^2	3
278CR56	Wustrowerstr. 15	13051	€829	€85	€85	71.2 m^2	3

Table 6 False Positives: Example 2

ID	Street	Postal Code	Rent Price	Utility Costs	Heating Costs	Living Area	Rooms Number
8265526	Rudolf-Seiffert-Str. 84a	10369	€588	€80	€53	102.2 m^2	2
26R945V	Rudolf-Seiffert-Str. 84a	10369	€571	€80	€53	38.03 m^2	2

Table 7 False Positives: Example 3

ID	Street	Postal Code	Rent Price	Utility Costs	Heating Costs	Living Area	Rooms Number
4803292	Kastanienallee 82	12627	€469	€51	€45	34.38 m^2	1
278CZ5B	Kastanienallee 48	12627	€459	€51	€44	34.38 m^2	1

Table 8 Evaluation of the Classifiers

Indicator	K-Means Classifier	EM-Classifier
Accuracy	0.7925	0.7720
Recall	0.9979	0.9979
Precision	0.7039	0.6838
Sensitivity	0.9979	0.9979
Specificity	0.5935	0.5531
F-score	0.8255	0.8115

Overall, one would recommend treating the number part of the address as a separate variable and not as a part of a lengthy string variable. For technical reasons, we did not perform such a split of the address information. We expect that ignoring differences in flat offers with almost the same address but more or less equal numerical variables is of minor importance for the flat rate.

If we compare the k-means classifier and the EM-classifier in Table 8 with respect to classical criteria the k-means performs slightly better. The biggest difference lies in the precision that indicates the quantity of corrected labeled positive cases from all labeled positive cases. The F-score which is just the trade between recall and precision yielded a good performance for both classifiers where the k-means classifier outperforms just for a small difference.

Under these conditions, both classifiers can be used accordingly. Hence, both methods were executed in parallel to compare the posterior results of the price prediction model. In the main document, the k-means approach will be presented, while the results from the other classifier can be found in the respective Jupyter Notebooks (Appendix C.4 and D.4) attached to this document.

Table 9 shows the number of repeated, unique, and total offers obtained in the merged dataset under each classifier. There is only a small difference in the quantities between the two schemes which still supports the similarity of performance of both of them. A detailed description of the record linkage and merging process can be found in the respective appended Jupyter Notebooks (Appendix C.1, C.2, D.1, and D.2) dedicated to these specific stages.

Table 9 Composition of the Merged Dataset

Classifier	Number of Duplicates	Number of Unique Offers	Total Number of Offers
K-means Classifier	1084	1866	2950
EM-Classifier	1074	1878	2952

5 Building a merged dataset

The output of the classification model is a list in which each element represents a comparison pair that was labeled as a match. Therefore, each element contains the IDs of the two offers forming the comparison pair. This is the basis of a consolidated dataset of the two portals.

Each real estate portal has its own set of variables, and those are defined according to the rules and formats of the respective online portals. Some variables have different names and attributes, and some that are present in one portal are missing in the other. Hence, if a merged dataset is going to be built, it should not be dominated by the information from only one of the two data sources. Furthermore, variables not common between the portals may lead to missing values, depending on the information available from the other portal. Additionally, common variables need to be synchronized. The common variables were identified, and renamed, and their attributes were standardized to ensure they conveyed the same meaning. In both datasets, there were groups of dummy variables related to specific features of the flat. For example, the presence of a garden, a balcony, or a terrace describes outdoor facilities that the flat might have. For these nominal variables with differing values, specific dummy variables were created to have identical meanings across both datasets. For example, a portal may mention the presence of a balcony and a loggia while the other just mentions the presence of a balcony. Thus, they can be combined into a single variable indicating the presence of either a balcony or a loggia.

After such synchronization, we have to decide which observation enters the merged dataset. For this purpose, we compute for each pair the number of variables with valid information. The offer from the portal with the higher number of valid information is chosen. However, before dismissing the less informative offer, we checked the possibility of augmenting information on some variables from the less informative offer to the more informative offer. We had also to consider the case that an offer appears in more than one pair. Here we used an ad hoc rule by keeping the newest offer.

The merged data frame contained originally 74 variables (See Table 12 and Juypter Notebook Appendix C.2 and D.2). To be able to come up with a smaller set of plausible and relevant variables some preprocessing steps were performed. In this procedure all variables with more than 50% of missingness, unbalanced categorical variables, and variables with possible multi-collinearity problems were removed.

Specifically, we mention the case of the variable “Deposit for the Rent”. The variable is an excellent predictor for the rent. So, every automatic variable search routine would use the Deposit variable to predict the rent. However, from subject matter knowledge it is known that the deposit is approximately equal to two to three times the monthly rent. This creates a substantial collinearity in the estimation of the regression coefficients for the flat rate per m^2. Thus results of automatic variable selection routines should be checked against subject matter knowledge.

Our routine left only eight variables for further analysis: Utility costs, Heating costs, number of rooms, living area, construction year, central heating, use of gas, and use of district heating. The whole process of variable screening can be seen in detail in the internet supplement (Jupyter Notebook Appendix C.2 and D.2).

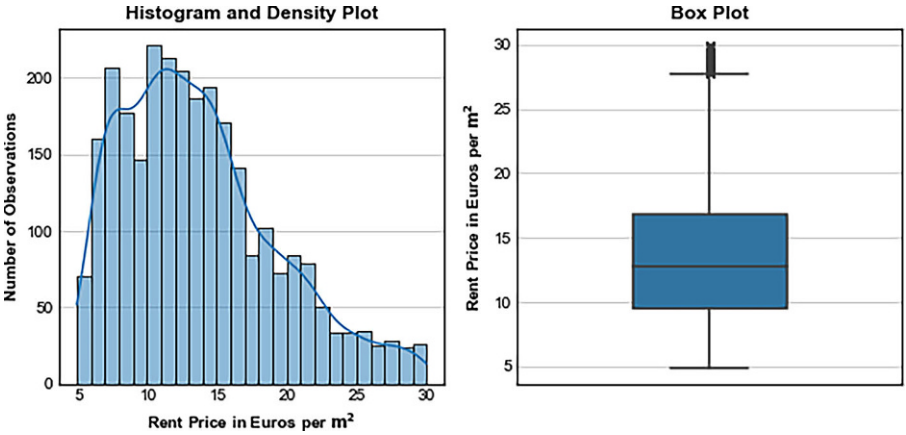

Fig. 2 Rent Price in Euros per m^2

5.1 Analysis of the merged dataset

A descriptive summary of the variables of the merged data set can be found in Table 13.

5.1.1 The distributions of the flat rate

The most important variable of our analysis is the rent per square meter. For this reason, we will display here its distribution in the merged dataset.

Fig. 2 displays a histogram and a density plot (left) and a box plot (right) of the rent price. The distribution is heavily right skew and the histogram displays truncations below 5 and above 35 Euros. Values beyond these limits were considered implausible and hence, they were deleted from the dataset. The mean of the rent prices in this range is 13.66 Euros. This value is far above the value of the official rent index for Berlin[1] which is 7.16 Euro. By law, this value refers to flats constructed not later than 2017. Thus the flats constructed between 2018 and 2022, which is the year of our web-scraping, are not included in the so-called "Mietspiegel". Even if we add an estimated increase of 4.40 Euro for the period 2017 to 2022 we are still below the average online flat rate[2]. There are two possible causes for this discrepancy. First, newly-built flats are more expensive than older flats. This is well documented for the periods before 2017 in the Mietspiegel table[3]. There, the average flat rates in the building class 2003 to 2017 are in the range of 11 to 13 Euros per m^2, which is far above the general average of 7.16 Euros. Thus, if the majority of the online offers

[1] See https://www.morgenpost.de/berlin/article238691783/mietspiegel-berlin-2023-ortsuebliche-vergleichsmiete-mieterhoehung.html (Access 5. Feb. 2024).

[2] The estimated increase of rent prices in Berlin from 2017 to 2022 is about 4.40 Euro, see https://de.statista.com/statistik/daten/studie/535119/umfrage/mietpreise-auf-dem-wohnungsmarkt-in-berlin/ (Access 5.Feb. 2024).

[3] https://www.stadtentwicklung.berlin.de/wohnen/mietspiegel/de/download/Mietspiegeltabelle2023.pdf (Access 5.Feb. 2024).

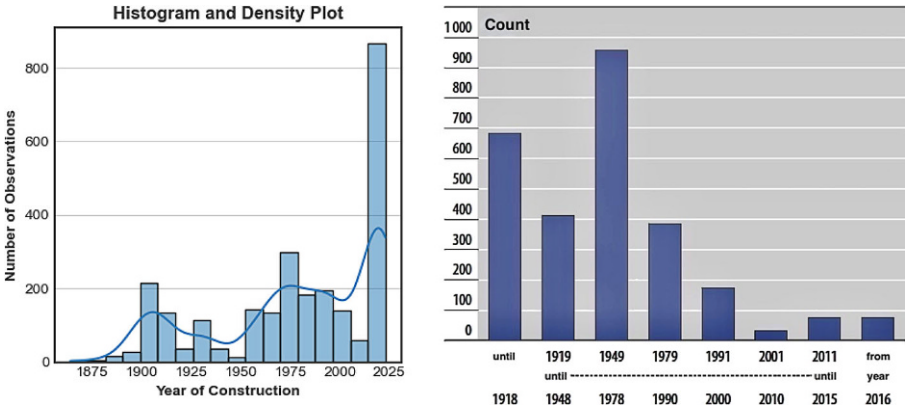

Fig. 3 Year of Construction. Left: Online Offers 2022 Right: Microcensus 2018

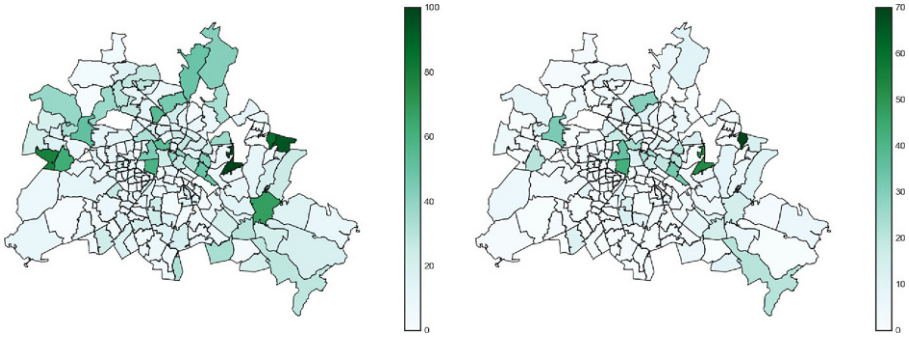

Fig. 4 Distribution of the Online Offers in Berlin Left: Number of Online Offers in ZIP Areas Right: Percentage of flats constructed beyond 2000 among online offers

refer to newly built flats this would explain the difference between the official rent index prices and the online offers.

Fig. 3 displays the year of construction of the flats in Berlin. The left panel displays the distribution of the online offers while the right panel documents the corresponding distribution of the data from Berlin (Frink and Rendtel 2019). In the representative microcensus distribution, most of the flats were built before the German re-unification in 1990, while the online offers of 2022 are mainly constructed after the year 2000.

Also, the location of the online offers is atypical for the stock of flats in Berlin which is situated in the central parts of the town. Fig. 4 displays in the left panel the number of online offers in the respective ZIP areas. They are primarily concentrated in the outskirts of the town. The exceptions are some central areas with a change of use, such as former industrial areas that were converted to housing areas. An example is the area near the newly-built Berlin main railway station[4]. As a rule, the

[4] For example, the so-called Europa-City with 3000 flats, https://europacity-berlin.de (Access 5.Feb. 2024).

percentage of offers built after 2000 is very high in these areas, see the right panel of Fig. 4.

All in all, the online market of flat offers is quite different from the stock of existing flat contracts. The standard survey instrument of official statistics, the subsample of the microcensus on flats which is taken every four years is not fast enough to track this special market. Also, the official rent index which is updated every two years does not cover this part of the flat market, because newly-built flats are excluded by law. Besides, there are some concerns about the representativeness of the Berlin rent index, see Rendtel et al. (2021).

5.1.2 The impact of the covariates on the flat rate via scatter plot smoother

Before we proceed to model the impact of the remaining covariates on the flat rate we should have a rough idea about the functional relationship of the covariate and the flat rate. We use here the two-dimensional scatter plot smoother which can be interpreted as a non-parametric estimate of the conditional expectation of the dependent variable, here the flat rate, given the value of the variable at the x-axis. This gives a hint about the polynomial degree of the impact of the covariate in a main-effects regression setting.

This knowledge is necessary due to a technical problem: Not all relevant variables which explain the flat rate are complete. Table 13 in the appendix displays standard descriptive statistics and also the percentage of missing data for each variable. For some variables, like heating costs, the percentage of missing data amounts up to 40 percent. If we apply an imputation strategy, like the MICE algorithm, knowing which variables to include in the imputation procedure is important. Here, it is the quadratic and cubic terms of the variable which may be relevant. Therefore, it is important to have an idea of the functional impact, for example, of the size of the living area on the flat rate. If the relationship is approximately cubic we have to

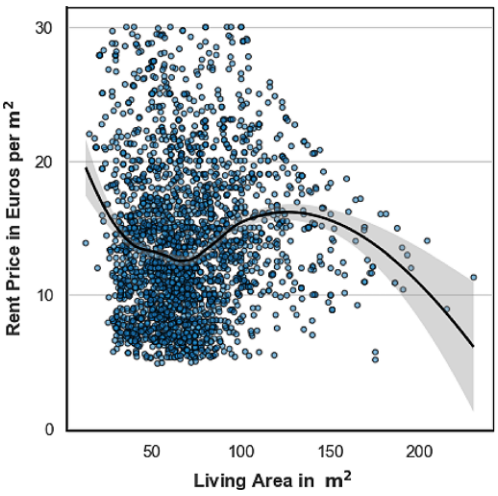

Fig. 5 Flat rate vs living area

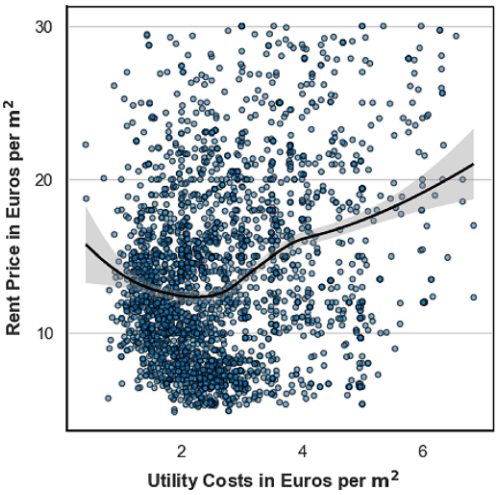

Fig. 6 Flat rate vs utility costs

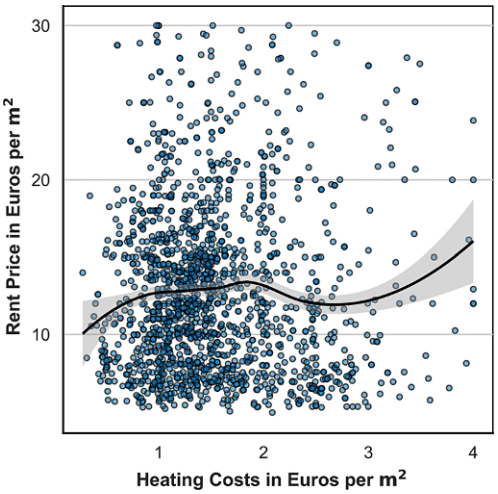

Fig. 7 Flat rate vs heating costs

impute not only missing values of the living area but also their squares and their cubic values, see Sect. 6 below.

Figs. 5 through 8 display the functional relationship. Due to the high number of observations, a mere scatter plot is not adequate for this purpose because they often condense into clusters where no functional relationship is visible, as is the case here. However, the scatter plot smoother which is also displayed in the scatter plots clearly exhibits the functional relationship.

It is well known that increases in the flat's living area result in decreased flat rates per square meter. However, Fig. 5 demonstrates that this trend is reversed for larger flats of about $100\,m^2$. Thus, we need at least a cubic polynomial to model the impact of the size of the living area on the flat rate. If we proceed to the utility costs per square meter in Fig. 6, we observe an approximate quadratic impact of

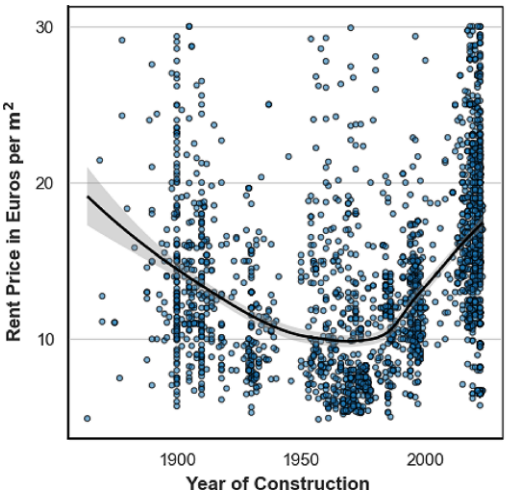

Fig. 8 Flat rate vs year of construction

this variable on the flat rate. There is almost no impact on the heating costs per m^2 (Fig. 7). Analogously, similar results are valid for the number of rooms. Finally, the year of construction (Fig. 8) indicates a strong quadratic impact on the flat rate. This indicates that houses, which were built in the fifties and sixties of the last century with a generally lower standard of living are cheaper than flats in buildings that were constructed before the second world war or are built after the seventies.

6 The treatment of missing values by multiple imputation

As the flat offers are based on free-formulated texts, there occurs non-response with a higher rate than in a structured questionnaire. This missingness in the data must be accounted for and corrected to yield valid inferences from the observed data. Here, we use multiple imputation to cope with missingness (van Buuren 2018).

Multiple imputation creates m versions of imputed datasets by replacing the missing values with possible plausible values. The plausible values are drawn from a modeled distribution for each specific entry or variable. Each imputed dataset is different concerning the imputed values reflecting the uncertainty inherent to them. After the imputation, we estimate the parameters of interest from each imputed dataset and pool them into a single estimate using Rubin's rules (Little and Rubin 2019). The variance of the estimate combines the sampling variance (within-imputation variance) and the additional variance caused by the missing data (between-imputation variance). Under appropriate conditions, the pooled estimators are unbiased and have the correct statistical properties (Little and Rubin 2019).

The MICE (Multiple Imputation Conditional Expectation) Package generates sequentially multiple imputations for missing values of the basic variables and their transformations, see van Buuren (2018). It starts with a prediction of the variable with the lowest rate of missingness. Its missing values are predicted using the vari-

Table 10 Used Single Imputation Methods in the MICE Package

Type of Variable	Single Imputation Method
Numeric	Predictive Mean Matching (PMM)
Categorical (2 levels)	Logistic Regression (Logit Regression)
Numeric Quadratic Term	Quadratic and Passive Imputation
Transformed or Derived Variable	Passive Imputation

ables that are complete. The type of prediction depends on the type of the variable. At this stage single imputation methods are used, see Table 10.

So step by step advancing to variables with increasing rate of missingness all missing values are imputed which marks the end of the initial phase of the MICE algorithm. Then the procedure is repeated by using the most recent imputations as predictors. These steps are repeated 50 times. Every 10th realization is then used for imputation, leading to a total of m=5 multiple imputations.

To check the quality of the imputations among the five imputed datasets, a density plot was generated for each variable as shown in Fig. 9. The blue lines represent the distribution of the observed data for each variable and the red lines represent the distribution of the imputed data for each version of the dataset. It can be seen that the distribution of the imputed data closely represents the distribution of the observed data in most cases. However, in the case of flat rates with only 6 percent

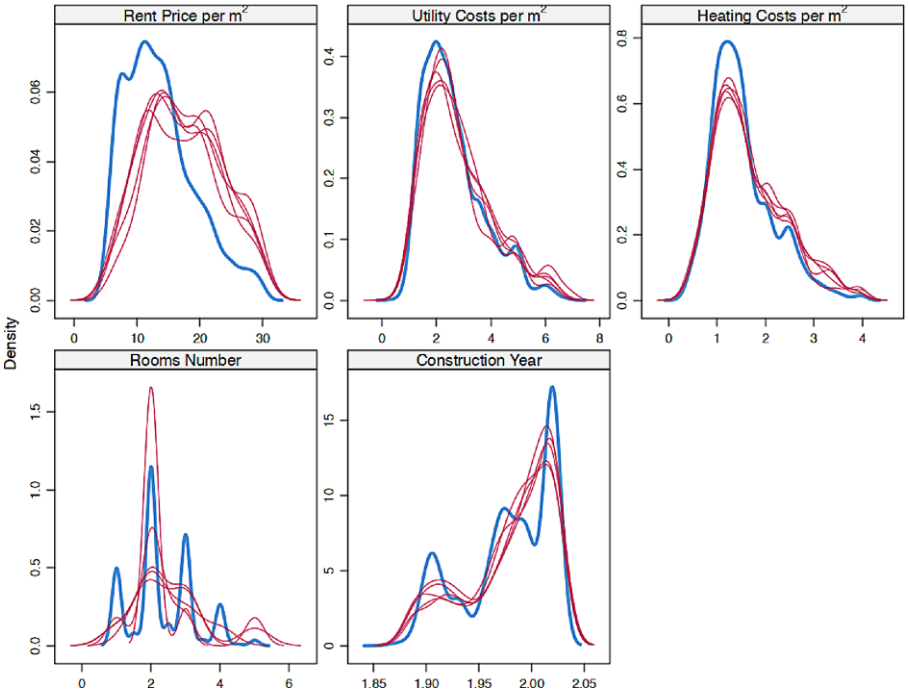

Fig. 9 Density Plot of Observed and Imputed Data: Thick blue line for observed data. Thin red lines for Imputed data

of missing values, the distribution of the imputed data is shifted to the right of the observed values. We conclude that missing values appear more frequently for flats with above-average prices.

7 Modeling

Here we first start with a variable selection procedure in a standard regression setting, an approach that is straightforward to implement. With the selected variables we switch to more demanding regression models that include geographical information.

7.1 Variable selection under OLS regression

After setting up and applying the MICE model, five different imputed datasets were generated. It is still an open statistical problem how variable selection can be done with multiply imputed data. With a fixed database there are well-established variable selection procedures; however, with $m = 5$ datasets, the search routine may yield a different model choice for each dataset.

For our start of the variable selection, we use ordinary least squares (OLS) regression where the coefficients and standard errors are pooled using Rubin's rules of multiple imputation. Here, an OLS model for each imputed dataset is estimated and then the coefficients and standard errors of the models are pooled by using these rules. The significance, standard errors, and the correlation matrix of the coefficients are analyzed and evaluated. According to these results, non-significant variables are dropped such that the fitted model improves. The R^2 and adjusted R^2 coefficients are evaluated as well. This is done until a first basic and plausible model is reached (see Table 14). To check for heteroscedasticity, the residuals plot of each imputed model was checked and a Breusch-Pagan test was used as well (Heij et al. 2004). For the variance estimates we used heteroskedasticity adjusted standard errors[5].

The obtained estimates are given in Table 15 in column "Model a". We then discard the variable with the highest p-values. The final model selection is "Model c" of Table 15. The parameter estimates reaffirm the findings from the two-dimensional scatter plots of the previous section. The heating costs per m^2 and the number of rooms have no impact on the flat rate. The impact of the living area is cubic while the impact of the construction year is quadratic. Furthermore, there is an impact on the energy source of the flat: central heating results in cheaper flat rates while gas heating implies higher flat rates. The adjusted R^2 of 0.310 is low and does not vary across the different models. However, it indicates that important sources of variance are not detected by this simple model.

A standard variable in the official rent index is the quality of the surrounding neighborhood of the flat. For this purpose, Berlin was separated into three different

[5] We used for our application the default HC2 standard errors (Hayes and Cai 2007; Blair et al. 2022) (Access 5.Feb. 2024).

quality groups: simple, medium, and good neighborhoods[6]. However, the area system of the quality districts is not compatible with the ZIP areas. Besides, the impact of the quality differences on flat rates is quite limited, as Frink and Rendtel (2019) have shown for rent rates from the microcensus. Instead, they found large effects of the ZIP areas on the flat rate.

After this phase of variable selection, we include spatial effects associated with the ZIP codes and allow a non-parametric impact of the living area on the flat rate.

7.2 Semiparametric models with spatial components

The linear regression model for a metric dependent variable can be extended in various ways ranging from Generalized Additive Models (GAM), Generalized Additive Mixed models (GAMM), Generalized Geoadditive Mixed Models (GGAMM), dynamic models, varying coefficient models, and geographically weighted regression. This broad range of Structured Additive Regression (STAR) models can be treated in a unified framework by the standalone Package BayesX (Belitz et al. 2015a). This package provides numerically efficient implementations via MCMC algorithms. In order to integrate BayesX and its results into the R-interface we used the package R2BayesX.

Our final model is displayed in Eq. (1):

$$\begin{aligned}\text{Rent Price} = {} & \beta_0 + (\text{Utility Costs})\beta_1 + (\text{Utility Costs})^2\beta_2 + f_1(\text{Living Area}) \\ & + f_2(\text{Postal Code}) + (\text{Construction Year})\beta_3 + (\text{Construction Year})^2\beta_4 \\ & + (\text{Central Heating})\beta_5 + (\text{Gas})\beta_6 + \text{Error}\end{aligned} \tag{1}$$

Here $f_1(x)$ displays the impact of the living area on the flat rate. $f_2(x)$ represents the regional effect, i.e. the ZIP code, on the flat rate. Here, the ZIP code can be treated either as variable at the nominal level or at the metric level where the geocoordinates of the ZIP-centriods are used.

In the first case, at the nominal level, a Markov field approach is used. A Markov field is a two-dimensional extension of the well-known one-dimensional Markov Chain. Here, only geographically neighboring regions, here ZIP areas, interact with each other. In this case the geographical coordinates of each observation are not required. If a smoothed spatial effect is desired, a spline approach can be used instead (Belitz et al. 2015b). Here, we use the centroids of the ZIP regions as coordinates.

The estimation of this model relies on the use of Markov Chain Monte Carlo techniques. Here a predefined quantity of first iterations, called the burn-in phase, is ignored. The estimates from later iterations are considered to be in the steady state distribution of the Markov chain. In this implementation, the default quantities of 2000 (burn-in) and 12000 (estimation period) iterations were used. Results are

[6] See https://www.stadtentwicklung.berlin.de/wohnen/mietspiegel/de/wohnlagenkarte.shtml (Access 5.Feb. 2024).

Table 11 Parametric Estimates of the non-parametric model

Variable	Mean	Sd	2.5%	50%	97.5%
(Intercept)	12.1482	0.5771	11.0315	12.1527	13.2578
Utility Costs	0.2113	0.3150	−0.4057	0.2206	0.8071
Utility Costs Squared	0.1395	0.0467	0.0499	0.1390	0.2342
Central Heating	−0.8758	0.1887	−1.2413	−0.8768	−0.5037
Construction Year	0.1137	0.0049	0.1039	0.1137	0.1235
Construction Year Squared	0.0011	0.0001	0.0010	0.0011	0.0012
Gas	0.2683	0.1718	−0.0624	0.2786	0.5923

presented for one imputed dataset. The overall behavior of these results can be generalized for the remaining four imputed models and can be seen in detail in the internet supplement (Jupyter Notebook, Appendix C7).

The parametric estimates of these models are shown in Table 11. It can be seen that the coefficients of the linear part of the model (which are represented by the mean column) are similar to those obtained under OLS estimation. However, the effect size is somewhat smaller. This implies that a part of this effect is now attributed to the spatial effect. The value of the intercept is due to a different scaling of the variables. The standard deviation and the respective quantiles of the estimation result from the 12000 replications of the algorithm. Here every 10th estimate was selected to replicate the distribution of the estimate.

Fig. 10 presents the estimated non-parametric effect of the size on the flat rate. The smooth function follows the same pattern as indicated in the scatter plot smoother in Fig. 5. We conclude that there is a price shift for larger flats, which decreases again as the flat size increases further.

The spatial effect is displayed in Fig. 11. The left panel shows the result for the Markov field specification, where only neighboring areas affect each other. However, the number of cases in the different ZIP areas varies substantially. There are even areas without any offer, which are left blank in Fig. 11. This reflects the fact that the large majority of offers are located in newly built houses which are distributed irregularly across the town.

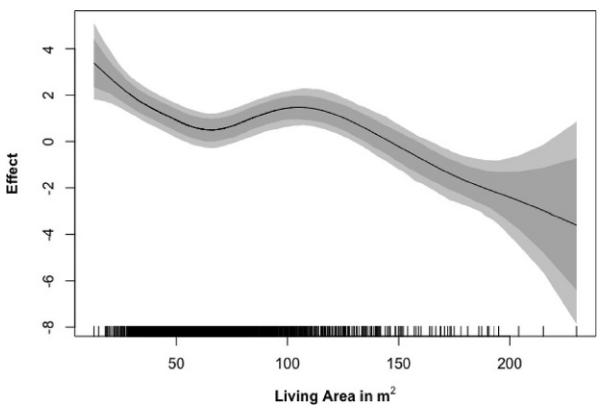

Fig. 10 Non-Parametric Effect of the Living Area with credibility intervals

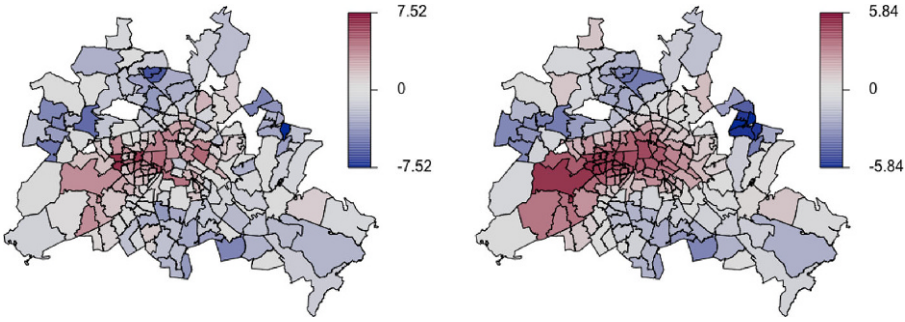

Fig. 11 Spatial effect on flat rate: Left: Markov Fields Right: Spline Approximation

To get a smoother regional effect we use the splines option in the BayesX package. The result is displayed in the right panel of Fig. 11. The magnitude of the spatial effect on the flat rate remains remarkable where the price differential amounts to 11 Euros per square meter!

As expected, the neighborhoods located in the central areas of the city tend to have the highest prices while the ones in the outskirts of the city tend to be cheaper with some exceptions. Some outskirts of the city are characterized by tower buildings from the 70s and 80s of the last century. This kind of construction is known in the German language as "Plattenbau" which refers to buildings with large, prefabricated concrete slabs. The specific exceptions in this area are neighborhoods with a traditional and exclusive background that makes them more expensive despite their location.

The last step is to produce pooled predictions to account for the uncertainty of the imputations. In the first stage, the fitted values of each model are calculated and then, they are pooled using Rubin's rules. Due to the complexity of the model, a simpler approach may be preferable. Here, a prediction is estimated by pooling the prediction of each imputed model instead.

Coefficient convergence analysis for the non-parametric and the parametric estimates of the model showed no specific trend along the iterations of the model and the resulting behavior for each coefficient was akin to white noise. This allows the assumption of convergence of the estimation of the model. These plots of the convergence of the coefficients can be checked in the internet supplement (Jupyter Notebook Appendix C7).

8 Conclusions

Web scraping has proven to be a powerful tool for the collection of a huge amount of data from different internet portals. However, one should be careful with the obtained data. Depending on the source, the information can be structured or unstructured which can cause further effort to develop an adequate scraper.

The production of a prediction model for flat rates using web scraping from different information sources was possible thanks to an assembly of a set of interconnected

steps with a specific objective along the whole process: A *record linkage model* to consolidate information from different sources, a *multiple imputation model* to account for the uncertainty caused by the missing data, *statistical methods for variable selection* that are included in the final model and finally, a *structured additive model* that allows accounting for the spatial effect of the postal codes of the offers and gives the flexibility of a semi-parametric model. This is a long lane that started from very technical issues of web scraping and ended with the use of advanced statistical routines. Needless to say a pure technical approach or an automatic treatment will not succeed in a satisfactory result.

Besides, we have encountered some problems with the representativeness of our results which are worth mentioning.

We could show that scraping data from the two portals, Immowelt and Immonet, leads to a substantial over-estimation of the number of offers. Even more intriguing is the fact that the offers that appear in both portals are more expensive than offers that appear only in one portal.

This might advocate for the use of only one portal. In this case, one could save the de-duplication via statistical matching. This might also reduce the missing data problem to some extent which results from different rules to document the flat offer. On the other hand, the omission of an important portal will under-count the number of flat offers. A major point of concern is whether the restriction to one portal is selective with respect to the estimation of a model for the flat rate. In the case of the two portals of our analysis, we found considerable differences with respect to the average flat rate. Hence we recommend scraping from both portals and also de-duplication of joint offers.

Such issues open the general question of the representativeness of results from online portals. In the context of flat offers, there will be probably low-price offers that do not appear in the Internet. Or they are offered in the internet but on portals that are small or not open for scraping. On the other hand, if we scrape more than two portals, the de-duplication or even de-triplification can become a challenging task.

Nevertheless, similar problems occur also in other contexts, for example, the scraping of job offers. As long as the population frame is not fixed, the problem of representativeness cannot be addressed. Since the entire market of local flat offers is hard to observe it might be regarded as sufficient to restrict the population frame to "scrapable" online portals.

More attention should be also spent on the sampling scheme from the portal. If we sample all offers at a given point in time then we will miss those offers which stay only for a short time interval on the Internet. This is mainly due to the fact that low-cost offers quickly find a tenant and were removed from the portal. Thus the more expensive offers remain on the Internet longer and are therefore over-represented by scraping the stock of offers, see Kauermann et al. (2021) on page 308 for an example. Such selective effects can be only avoided by a daily scraping of new offers. Still, with respect to the before-mentioned difficulties of de-duplication, the daily selection of new offers may be too tedious.

A topic that was not discussed here is the presence of measurement errors in data from online portals. The paid rent consists of three components: the costs for heating,

the utility costs, and the share for the landlord (In German: “Nettokaltmiete”). Only the last component is relevant for rent indices. Nonetheless, it may happen that the total rent or the rent plus the utility costs are given in the exposé. It depends on the instructions of the portal and how well these components are separated. Thus, there is the tendency that the flat rates from the online exposés over-estimate the true rent. Here the questionnaire-based surveys of the local rent index and the microcensus may deliver more accurate results.

Our prediction model indicates substantial and plausible regional effects on the flat rate. These effects are in line with corresponding Berlin results from the German microcensus in Frink and Rendtel (2019).

As we have shown, the online market for flats is quite different from the stock of all rent contracts which is the focus of local rent indices. The online offers are concentrated on newly-built houses in outskirt areas of Berlin. Also, the price level is far above the Berlin rent index. This raises doubts about the ability of the local rent index to judge a flat rate of a new contract as “fair”.

Despite these problems, rent price models derived from online offers can be a helpful tool for official statistics to monitor the temporal development of flat rates. Here, the 4-year time interval of the German microcensus is too long to keep track of the actual rapid increase of flat rates in Germany.

9 Appendix

The appendices related to this document consists on a series of Jupyter notebooks containing a step by step procedure of each stage of the project. All programmed code and results can be found there respectively. Because of their large size we have stored them in a publicly available archive under the following URL https://github.com/meybergc/berlin_webscrapping. Each Jupyter notebook is presented there as a “html” file. The files can be downloaded and opened by any browser. The list of the appended Jupyter Notebooks is given as follows:

A Diagnostic checking and preprocessing of Immowelt
B Diagnostic checking and preprocessing of Immonet
C K-means classifier linked data

C.1 Record – Linkage
C.2 Generating a merged dataset
C.3 Quick look to the merged dataset
C.4 Multiple imputation model
C.5 Ordinary Least Squares
C.6 Ordinary Least Squares with heteroskedasticity-robust standard errors (HCSE)
C.7 Structured Additive Models

Table 12 Scraped Variables and Preprocessing of the Data

Variable/Group Name	Preprocessing Approach	Real Estate Portal	
		Immonet	Immowelt
Postal Code	– Check the plausibility of the available ones using geolocation – Remove offers with postal codes not correspondent to Berlin	Yes	Yes
Net Rent Price (in Euros)	– Remove string characters and turn the variable into a float – Remove offers which values are higher than €3000 Euros – Remove offers which values are higher than €30 and less than €2 per m^2 – Calculate missing values where related variables are available	Yes	Yes
Gross/Total Rent Price (in Euros)	– Remove string characters and turn the variable into a float – (Immowelt) Remove offers which values are higher than €5000 – (Immowelt) Remove offers which values are higher than €40 per m^2. This observations did not have information about the rental price nor utility costs available. Too high values with respect to the size of the flat – Calculate missing values where related variables are available	Yes	Yes
Inclusion of the heating costs in the total rent price (Dummy)	– Assume that missing values means that the heating costs are not included in the total rent price – Adjust total rent price with implausibilities where the heating costs are included	Yes	No
Utility Costs (in Euros)	– Remove string characters and turn the variable into a float – Remove offers which values are higher than €7 per m^2 – Remove offers with zero or negative values – Calculate missing values where related variables are available	Yes	Yes
Heating Costs (in Euros)	– Remove string characters and turn the variable into a float – Remove offers which values are higher than €4 per m^2 – Calculate missing values where related variables are available – Remove offers with negative values or close to zero where the heating costs are included in the total rent price. It means, they are already included in the utility costs	Yes	Yes
Region	– (Immowelt) Complete the missing values and check the plausibility of the available ones using geolocation – (Immonet) Join the scrapped offers to the region assigned on the search results of the portal using the url – Filter offers from Berlin	Yes	Yes

Table 12 (Continued)

Variable/Group Name	Preprocessing Approach	Real Estate Portal	
		Immonet	Immowelt
Living Area (in m^2)	– Remove string characters and turn the variable into a float – Remove offers which values are higher than 250 m^2 – Remove offers where the living area is missing and there is not enough information available to impute	Yes	Yes
Deposit (in Euros)	– Remove string characters and turn the variable into a float – Use text processing to calculate the respective numeric value according to the string – Remove offers which values are higher than €10000	Yes	Yes
Energy Certificate	– Standardize the string values	Yes	Yes
Energy Efficiency Class	– Standardize the string values	Yes	Yes
Number of Parking lots	– Turn the variable into float – (Immowelt) Extract the numbers using text processing	Yes	Yes
Type of Flat	– (Immonet) Use text processing to extract the respective value according to the subtitle of the offer – (Immowelt) Two variables contained this information. One of them was selected and standardized	Yes	Yes
Rooms Number	– Turn the values into float – Remove offers with more than 5 rooms	Yes	Yes
Year of Construction	– Use text processing to extract the values and turn the values into integers – Extract the first available year in case that the offer contained more than one year. They refer to some works done to the construction – Not plausible values were considered as missing values – All flats constructed from 1850 were considered	Yes	Yes
Heating Type	– Split the variable into dummies. One flat can have more than one type at the same type and at least one.	Yes	Yes
Heating Energy Source	– Split the variable into dummies. One flat can have more than one source at the same type and at least one.	Yes	Yes
Conditions of the Flat	– Standardize the string values	Yes	Yes
Current Use of the Flat	– No preprocessing required	No	Yes

Table 12 (Continued)

Variable/Group Name	Preprocessing Approach	Real Estate Portal	
		Immonet	Immowelt
Type of Flat	– (Immonet) Use text processing to extract the respective value according to the subtitle of the offer – (Immowelt) Two variables contained this information. One of them was selected and standardized	Yes	Yes
Availability of the Flat	– Use text processing to extract the values and split them into categories, years and dates – Complex strings were not considered	Yes	Yes
Energy Consumption	– Turn the variable into float	Yes	Yes
Additional Features	Groups of specific features of a flat. These groups were split in dummy variables. If at least one of the items of an specific group is available, the dummy variable correspondent to the present items are set to 1 while the others are set to zero. If no item in the group is present, all items in the group are set as missing values. This groups are the following:		
	– Type of parking	Yes	Yes
	– Furniture	Yes	Yes
	– Security facilities	Yes	Yes
	– Outdoors facilities	Yes	Yes
	– Wellness facilities	Yes	Yes
	– Sight features	Yes	Yes
	– Kitchen facilities	Yes	Yes
	– Bathroom facilities	Yes	Yes
	– Additional rooms	Yes	Yes
Additional Features	– Type of floor	Yes	Yes
	– Type of roof	Yes	Yes
	– Supply of the Building	No	Yes
	– Number of floors	Yes	Yes
	– Floor of the flat	No	Yes
	– Ventilation Appliances	Yes	Yes
	– Elevator	Yes	Yes
	– Windows features	No	Yes
	In Immonet, all the groups of variables are found in a single string line while in Immowelt, this categories are already identified in the JSON-file but for each group all atributes are identified in a single text line as well. It is important to mention that depending on the group of variables, a not mention of an specific item, does not implies that this specific item is not present in the offer. This leads to the fact that some dummy variables are not reliable nor plausible.		

Table 13 Summary of Descriptive Statistics

Variable	Units	Mean	STD	MIN	Q1	Q2	Q3	Max	% Missing Data
Living Area	m^2	69.1293	27.9109	13	50	65	83	230	0.0000
Rent Price	Euros	961.6157	584.5215	163.3100	519	789	1268.50	3000	6.0338
Rent Price per m^2	Euros/m^2	13.6694	5.57326	4.8620	9.5001	12.8233	16.8079	30	6.0338
Utility Costs	Euros	180.4601	103.6944	16.7500	103	155	227.51	900	9.9661
Utility Costs per m^2	Euros/m^2	2.6212	1.1377	0.4085	1.7667	2.3809	3.1827	6.8493	9.9661
Heating Costs	Euros	101.9628	56.2296	10	62	90	130	450	38.8474
Heating Costs per m^2	Euros/m^2	1.5176	0.6404	0.2759	1.0657	1.3699	1.8462	4	38.8474
Year of Construction	Year	1978	41.5474	1864	1956	1986	2019	2023	11.2881
Number of Rooms	Rooms	2.3296	0.9197	1	2	2	3	5	0.2711
Dummy	Units	%	–	–	–	–	Count of Zeros	Count of Ones	% Missing Data
Central Heating	0/1	0.7481	–	–	–	–	415	1233	44.1355
Gas	0/1	0.4250	–	–	–	–	1269	938	25.1864
District Heating	0/1	0.4644	–	–	–	–	1182	1025	25.1864

Table 14 OLS Models

Dependent Variable: Rent Price (Euros/m^2)						
	Model 1a	Model 1b	Model 1c	Model 2a	Model 2b	Model 2c
(Intercept)	16.95829***	16.68078***	16.73616***	15.15779***	15.06231***	15.15767***
(Euros/m^2)	(1.19242)	(1.13683)	(0.99931)	(1.22297)	(1.14968)	(1.03012)
	[1.25223]	[1.19550]	[1.07060]	[1.30535]	[1.23634]	[1.13066]
Utility Costs	1.16809***	1.17018***	1.17469***	1.14667***	1.14805***	1.15561***
(Euros/m^2)	(0.09899)	(0.09994)	(0.10668)	(0.10450)	(0.10501)	(0.11452)
	[0.11010]	[0.11080]	[0.11711]	[0.11523]	[0.11547]	[0.12422]
Heating Costs	0.02527	0.03676		0.05746	0.06150	
(Euros/m^2)	(0.24746)	(0.24842)		(0.26015)	(0.26094)	
	[0.25232]	[0.25279]		[0.26389]	[0.26437]	
Rooms Number	0.03787			0.00744		
	(0.17866)			(0.18170)		
	[0.16837]			[0.17230]		
Living Area	−0.21135***	−0.20927***	−0.20962***	−0.21582***	−0.21547***	−0.21604***
(m^2)	(0.03704)	(0.03514)	(0.03442)	(0.03808)	(0.03618)	(0.03544)
	[0.03955]	[0.03839]	[0.03784]	[0.04227]	[0.04109]	[0.04057]
Squared Living Area	0.00230***	0.00230***	0.00230***	0.00246***	0.00246***	0.00246***
	(0.00039)	(0.00038)	(0.00038)	(0.00040)	(0.00039)	(0.00039)
	[0.00044]	[0.00044]	[0.00043]	[0.00047]	[0.00047]	[0.00047]

Table 14 (Continued)

Dependent Variable: Rent Price (Euros/m^2)						
	Model 1a	Model 1b	Model 1c	Model 2a	Model 2b	Model 2c
Cubic Living Area	−0.00001***	−0.00001***	−0.00001***	−0.00001***	−0.00001***	−0.00001***
	(0.00000)	(0.00000)	(0.00000)	(0.00000)	(0.00000)	(0.00000)
	[0.00000]	[0.00000]	[0.00000]	[0.00000]	[0.00000]	]0.00000]
Construction Year	0.13432***	0.13361***	0.13360***			
(Transformation: Construction Year – 2000)	(0.00582)	(0.00592)	(0.00595)			
	[0.00644]	[0.00656]	[0.00658]			
Squared Construction Year	0.00146***	0.00145***	0.00145***			
	(0.00006)	(0.00006)	(0.00006)			
	[0.00007]	[0.00007]	[0.00007]			
Central Heating	−1.41672*	−1.41190*	−1.40819*	−1.58114*	−1.57864*	−1.57515*
(Reference: Other Heating System)	(0.42181)	(0.42533)	(0.42246)	(0.45535)	(0.45422)	(0.45112)
	[0.41988]	[0.42331]	[0.42052]	[0.45228]	[0.45107]	[0.44804]
Gas	0.35386	0.54594*	0.54795*	0.82600*	0.89464***	0.89704***
(Reference: Other Energy Source)	(0.33463)	(0.22930)	(0.22411)	(0.34935)	(0.23895)	(0.23544)

Table 14 (Continued)

Dependent Variable: Rent Price (Euros/m^2)						
	Model 1a	Model 1b	Model 1c	Model 2a	Model 2b	Model 2c
	[0.32172]	[0.22072]	[0.21513]	[0.33785]	[0.23296]	[0.22920]
District Heating	−0.27349			−0.10107		
(Reference: Other Energy Source)	[0.29770]			[0.30814]		
	[0.28980]			[0.30010]		
New Construction				4.73382***	4.72389***	4.72128***
(Reference: Construction Year < 2000)				(0.26194)	(0.26444)	(0.26711)
				[0.27037]	[0.27429]	[0.27685]
R2	0.313	0.312	0.312	0.284	0.284	0.284
R2 Adj.	0.310	0.310	0.310	0.282	0.282	0.282
AIC	17533.6	17531.1	17530.9	17650.8	17647.3	17647.4
BIC	17611.5	17596.9	17590.8	17722.7	17707.2	17701.3

p-values reference: + $p < 0.1$, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

This table presents the selection of the explanatory variables that are relevant for the prediction of the rent prices per squared meter under two different schemes. All the models that begin with the number one are estimated using the quadratic relationship between the years of construction and the rent prices while the models that are denoted with the number two use a dummy variable that splits the constructions into new and old. The letters a,b,c indicate the stage of the selection process according to the scheme. For example, the models denoted with an a contain all the shown set of explanatory variables while b and c represent the estimated models after the not-significant variables in the previous models were dropped. The upper row of each explanatory variable contains the estimated pooled coefficient for each model, the middle row shows the calculated pooled standard errors denoted in parenthesis while the lower row represents the corresponding pooled white standard errors of the regression which are found within the brackets. In order to make the models under the two schemes comparable, the year of construction was transformed by subtracting the year of reference that the dummy variable uses to split the data. Then, the intercepts are brought to the same scale

Table 15 OLS estimates for the multiply imputed data base. Standard errors according to Rubins rule in brackets

Dependent Variable: Flat Rate (Euro/m^2)			
	Model a	Model b	Model c
(Intercept)	19.40555***	19.08611***	19.12490***
	[1.47237]	[1.41420]	[1.2326]
Utility Costs	−0.73736+	−0.71533+	−0.71293
(Euro/m^2)	[0.42616]	[0.42185]	[0.42977]
Squared Utility Costs	0.29515***	0.29225***	0.29236***
(Euro/m^2)	[0.06831]	[0.06770]	[0.06705]
Heating Costs	0.01158	0.02602	
(Euro/m^2)	[0.25393]	[0.25365]	
Rooms Number	−0.02126		
	[0.16840]		
Living Area	−0.20274***	−0.20408***	−0.20430***
(m^2)	[0.04081]	[0.03967]	[0.03907]
Squared Living Area	0.00225***	0.00226***	0.00226***
	[0.00045]	[0.00045]	[0.00045]
Cubic Living Area	−0.00001***	−0.00001***	−0.00001***
	[0.00000]	[0.00000]	[0.00000]
Construction Year	0.13549***	0.13456***	0.13456***
(Transformation: Construction Year – 2000)	[0.00639]	[0.00647]	[0.00650]
Squared Construction Year	0.00146***	0.00145***	0.00145***
	[0.00007]	[0.00007]	[0.00007]
Central Heating	−1.35902*	−1.35287*	−1.34914*
(Reference: Other Heating System)	[0.41079]	[0.41607]	[0.41312]
Gas	0.25570	0.48428*	0.48573*
(Reference: Other Energy Source)	[0.30939]	[0.21489]	[0.20948]
District Heating	−0.32036		
(Reference: Other Energy Source)	[0.28982]		
R^2 Adj.	0.320	0.320	0.319
AIC	17503.3	17501.3	17501.2
BIC	17587.2	17573.2	17567.0

p-values reference: + $p < 0.1$, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Funding Open Access funding enabled and organized by Projekt DEAL.

Conflict of interest The authors declare that they have no conflict of interest.

References

Agnello L, Schuknecht L (2011) Booms and busts in housing markets: determinants and implications. J Hous Econ 20(3):171–190

André C, Girouard N (2009) Housing market challenges in Europe and the United States. In: Housing markets, business cycles and economic policies. Palgrave Macmillan, S 109–130

Arce O, López-Salido D (2011) Housing Bubbles. Am Econ J Macroecon 3(1):212–241

Arregui N, Lariau A, Oman W (2009) Spain: selected issues. Technical report, international monetary fund (IMF). Country report no. 2022/046

Belitz C, Brezger A, Klein N, Kneib T, Lang S, Umlauf N (2015a) BayesX: software for Bayesian inference in structured additive regression models. Version 3.0.2

Belitz C, Brezger A, Klein N, Kneib T, Lang S, Umlauf N (2015b) BayesX: software for Bayesian inference in structured additive regression models. Methodology manual. Version 3.0.2. https://www.uni-goettingen.de/de/document/download/65a4cfe1e4c5de4e83959956bf50201b.pdf/methodology_manual.pdf. Zugegriffen: 2023-01-12

Bergenstrahle S (2016) The importance of affordable rental housing. Unpublished International Union of Tenants. https://www.iut.nu/wp-content/uploads/2019/02/SB_The_importance_of_affordable_rental_housing-2017.pdf. Zugegriffen: 2022-19-08

Blair G, Cooper J, Coppock A, Humphreys M, Sonnet L (2022) estimatr: fast estimators for design-based inference. https://declaredesign.org/r/estimatr/. Zugegriffen: 2023-01-14

Botchkarev A (2019) A new typology design of performance metrics to measure errors in machine learning regression algorithms. Interdiscip J Inf Knowl Manag 14:45–76. https://doi.org/10.28945/4184

Caldera A, Andrews D (2011) To move or not to move: what drives residential mobility rates in the OECD? OECD Economics Department Working Papers 846

Catte P, Girouard N, Price R, André C (2004) Housing markets, wealth and the business cycle. OECD economics department working papers no. 394. OECD Publishing

Christen P (2008) Febrl – A freely available record linkage system with a graphical user interface. In: Proceedings of the 14th ACM (Association for Computer Machinery) SIGKDD (Special Interest Group on Knowledge Discovery in Data) International Conference on Knowledge Discovery and Data Mining, S 1065–1068

Christen P (2012) Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection. Springer, Berlin Heidelberg

Collins M (2012) The naive Bayes model, maximum-likelihood estimation, and the EM algorithm. Columbia university. https://www.cs.columbia.edu/~mcollins/em.pdf. Zugegriffen: 2024-01-24

Czerniak A, Rubaszek M (2018) The size of the rental market and housing market fluctuations. Open Econ Rev 29:261–281

de Bruin J (2022a) Annotation. https://recordlinkage.readthedocs.io/en/latest/annotation.html. Zugegriffen: 2024-06-15

de Bruin J (2022b) Record linkage Toolkit documentation. Release 0.15. https://readthedocs.org/projects/recordlinkage/downloads/pdf/latest/. Zugegriffen: 2023-01-14

de Bruin J (2022c) Python record linkage toolkit documentation. https://recordlinkage.readthedocs.io/en/latest/index.html. Zugegriffen: 2022-24-08

Elasticsearch BV (2022a) The closer, the better. https://www.elastic.co/guide/en/elasticsearch/guide/current/decay-functions.html. Zugegriffen: 2023-01-14

Elasticsearch BV (2022b) What is Elasticsearch? https://www.elastic.co/what-is/elasticsearch. Zugegriffen: 2023-01-14

Elfeky M, Verykios V, Elmagarmid A (2002) 02. TAILOR: A record linkage toolbox, S 17–28

European Commission (2024) Web intelligence network. https://cros.ec.europa.eu/book-page/web-intelligence-network-project-overview. Zugegriffen: 2024-05-08

Frink N, Rendtel U (2019) Die Erhebung der Wohnungsmieten im Mikrozensus: Ein Instrument zur Validierung von Mietspiegeln? Z Amtliche Stat Berlin Brandenbg 4:48–65

Goutte C, Gaussier E (2005) 04. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation Bd. 3408, S 345–359

Hayes AF, Cai L (2007) Using heteroskedasticity-consistent standard error estimators in OLS regression: an introduction and software implementation. Behav Res Methods 39:709–722

Heij C, de Boer P, Franses P, Kloek T, van Dijk H (2004) Econometric methods with applications in business and economics. Oxford University Press

Kauermann G, Küchenhoff H, Heumann C (2021) Statistical foundations, resoning and inference for science and data science. Springer, Berlin Heidelberg, S 308

Kofner S (2014) The German housing system: fundamentally resilient? J Hous Built Environ 29(2):255–275
Leamer E (2007) Housing IS the business cycle. NBER Working Papers No. 13428. National Bureau of Economic Research
Little RJ, Rubin DB (2019) Statistical analysis with missing data. Wiley, S 793
Mamun AA, Aseltine R, Rajasekaran S (2016) Efficient record linkage algorithms using complete linkage clustering. PLoS ONE 11:e154446. https://doi.org/10.1371/journal.pone.0154446
Ong TC, Mannino MV, Schilling LM, Kahn MG (2014) Improving record linkage performance in the presence of missing linkage data. J Biomed Inform 52:43–54
Ricciato F, Wirthmann A, Hahn M (2020) Trusted smart statistics: how new data will change official statistics. Camb Univ Press Data Policy 2:e7
Rubaszek M, Rubio M (2020) Does the rental housing market stabilize the economy? A micro and macro perspective. Empir Econ 59:233–257
Schmandt M (2021) Zur Bezahlbarkeit von Wohnraum in Berlin. Z Amtliche Stat Berlin Brandenbg 3+4:66–73
Scrapy developers (2022a) Frequently asked questions. https://docs.scrapy.org/en/latest/faq.html. Zugegriffen: 2022-01-09
Scrapy developers (2022b) Spiders. https://docs.scrapy.org/en/latest/topics/spiders.html#scrapy.Spider.parse. Zugegriffen: 2022-01-09
Sebastian S, Memis H (2021) gif-Mietspiegelreport 2021. Auswertung der Mietspiegel der zweihundert größten Städte Deutschland. Technical report. https://digital.zlb.de/viewer/metadata/34340023_2021/1/LOG_0003/. Zugegriffen: 2023-08-09
Steorts R, Schmid T, Tzavidis N (2020) Smoothing and benchmarking for small area estimation. Int Stat Rev 88(3):580–598
van Buuren S (2018) Flexible imputation of missing data, 2. Aufl. Chapman and Hall/CRC