Rent Data Analysis and Prediction

FOR APARTMENTS IN GERMANY

Femi Onafalujo | Springboard Data Science | September 17, 2021

Executive Summary

This project confronted the task of helping landlords determine the rent of their apartments by developing a machine learning model that used features in a rental property dataset to predict rent. The aforementioned dataset was acquired from Immoscout24 that some consider to be the largest real estate website in Germany. Consisting of over 263,000 apartment-for-rent records with 49 descriptors including rent, number or rooms and the size of the apartment, the rentals dataset was instrumental in analyzing the factors affecting rent towards the development of machine learning model. After handling data integrity issues like missing data, outliers and inconsistent categorical values for categorical features, analysis of the dataset highlighted the living area, service charge, condition of the apartment and state in which the apartment resides as critical features in determining rent. On the basis of the results from exploratory data analysis, five main machine algorithms – to include linear regression, ridge, regression, lasso regression, random forest and XGBoost – were used to create models that predicted rent from the input features. Of these, the XGBoost model with an r-squared score of 0.866 and mean absolute error of €90.34 performed the best on an unseen test set. To further assess the performance of this model, the rent of 5 sample apartments were determined with the model, of which one out of the four apartments fell within the range of values predicted by the model. Although, not tried, there is optimism that the model would be useful in predicting rents on a larger sample size and other unforeseen rent records that any landlord may have. That said, a limitation of the model for future adoption is the time the dataset was collected, which was pre-COVID; COVID19 may have significantly altered the dynamics of the rental market to render the model suspect. Nonetheless, there is optimism that the same approach adopted in analyzing the rentals dataset, building, assessing, and selecting a machine learning model, would be applicable to a refreshed dataset.

Contents

Introduction

After investing much time, energy and money to prepare and stage an apartment for rent, the last thing a landlord wants is for the apartment to sit empty for a month, two months or more before the property clears. The rent of a property plays a crucial role in attracting a good number of high-potential prospective tenants, such that the property clears within a reasonable amount of time. In real estate, as in other industries, time is money, and the longer a property remains on the market, the worse-off the financial performance of the asset. For one, monthly expenses such as the mortgage, property tax and heat still require payment regardless of whether or not the apartment is occupied. Landlords are keenly aware of these concerns and strive to set the appropriate rent for a property.

PROJECT OBJECTIVE

This project explores the location, asset-specific, and macro-economic factors that influence the rent stipulated for an apartment in an effort to create a rudimentary model that can assist landlords in Germany to set their apartment rents.

The eventual model was developed to predict "base rent", which is the rent without additional costs, such as heating, electricity, internet, and cable costs.

PROJECT SCOPE

The Data Science Method (DSM) was used to progressively develop the project. DSM consists of data wrangling, exploratory data analysis, data pre-processing and model training, and model application. These steps will be described in the following sections.

DATA SOURCES

The primary dataset for our analysis was from Kaggle.ca. The data was originally scraped from Immoscout24 - the biggest real estate website in Germany. Immoscout24 lists both properties for rent and for sale, although only rental information was included with the dataset. The dataset (downloaded as a csv file) had over 268,000 rental listings, scraped on four days - September 30, 2018; May 31, 2019; October 31, 2019 and February 29, 2020. Each listing was described by 49 features comprising of location data, data on monthly expenses including rent, asset-specific data and time data. The Kaggle dataset can be accessed here (https://www.kaggle.com/corrieaar/apartment-rental-offers-in-germany). This dataset will be referred to as the 'rentals' dataset going forward.

The rentals dataset was supplemented by a state-wide macro-economic dataset extracted from Wikipedia. Each state was described by four main features to include its area, 2019 population, Human Development Index and GDP per capital in 2018. This dataset will be referred to as the 'state-features' dataset going forward.

Data Wrangling

The data wrangling process required the collection, organization and cleaning of data to allow for proper rent analysis and prediction. Ultimately, the intent was to create a final dataset that was reflective of the ground truth. Acquiring data was relatively straightforward and consisted of importing the Kaggle csv file, as well as importing the Wikipedia dataset to the notebook from the related web link. Cleaning the data was more challenging, particularly the rentals dataset. Missing values, features with many categories, text features, outliers, duplicate entries, inaccurate cross-field validation of fields, inconsistent category names, and data leaks, were some of the challenges confronted and resolved during the data cleaning process. A few of these issues are discussed in the following sections; however, a complete treatment of the data can be found in the data wrangling notebook. The original 268,850 observations with 49 features of the rentals dataset were trimmed down to 267,859 observations and 30 features after the data cleaning process was complete.

DEALING WITH MISSING VALUES

Only the rentals dataset had features with missing values, which were resolved on a feature-by-feature basis. Of the 49 original features, over half of them had missing values, with some having over 80% of their values missing. In total, 27 features had missing values, with 13 of these features having a numerical data type, 12 having a categorical data type and two having a datetime datatype. The figure below shows the 20 features with the most missing values, where the red line indicates a threshold below which over 40% of values for a feature are missing.

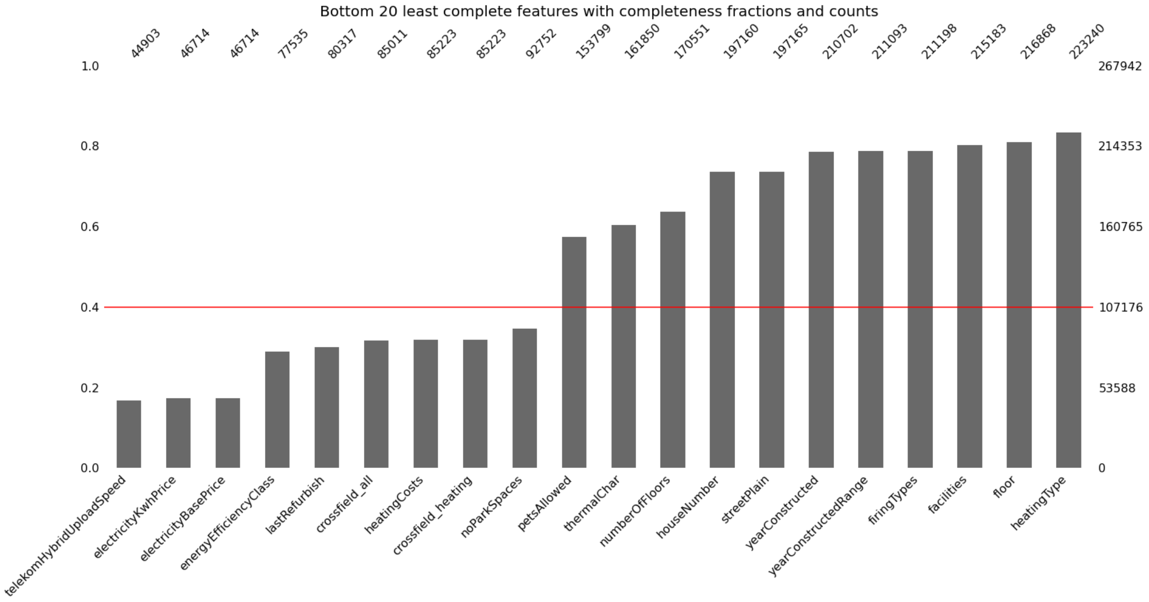


Figure 1: Bottom least complete features with completeness fraction and count

The general approach to dealing with the missing values of categorical features was to first of all assess the importance of the feature and then impute missing values with ‘unknown’ if the feature was deemed to be important. Research was essential in understanding why some features had many missing values. For instance, the energy efficiency class feature had over 70% of its values missing because certification requirements were a new policy mandate recently ushered in by the German government, and many landlords had not conducted an energy assessment to include this information in their listing. This feature was dropped from the dataset.

Similar to the categorical features, numerical features with missing values were reviewed on a case-by-case basis for their importance and kept for imputation with their median value if the feature was deemed to be important. Median imputation was preferred because of the presence of outliers in most of the numerical features

Importance was assessed by reviewing the number of unique values, and the standard deviation of these values. Features with few unique values and small standard deviations were considered unimportant and dropped from the dataset. For instance, the telekomHybridUploadSpeed feature had over 80% of its values missing and had one unique value (10.00). This suggested that this feature was not important.

REDUNDANT COLUMNS AND DUPLICATE ENTRIES

The rentals dataset had some observations and features repeated, which artificially increased the size of the dataset. Duplicated observations were verified to be repeated and dropped from the data set. About 890 observations were repeated and were dealt with accordingly.

The duplication of some features was due to a change in format. For example, street and streetPlain were the same feature, with one feature having special German characters like ‘ß’ converted to ‘&szlig;’ as a possible English alternative.

Other duplicated features were just given different names, which was unknown. For example, regio1 and geo_bln (representing the state name) were the exact same feature. One of the features was dropped from the dataset.

INCONSISTENT CATEGORICAL VALUES

The state-features dataset had different format for state names than the rentals dataset. For example, Sachsen_Anhalt in the rentals dataset was called Saxony-Anhalt (Sachsen-Anhalt) in the state-features features dataset. This particular issue was resolved using a string similarity algorithm to change the state names in the state-features dataset to the names in the rentals dataset.

OUTLIER DETECTION AND REMOVAL

The rentals dataset was such that the distributions of most of the numerical features were skewed, suggesting the presence of outliers. The typical bounds of a certain parameter were determined by research and the outliers filtered out of the dataset. For instance, the number of floors feature had had values up to 999, which was erroneous because the tallest building in Germany was determined to be the Commerzbank Tower in Frankfort with 56 floors.

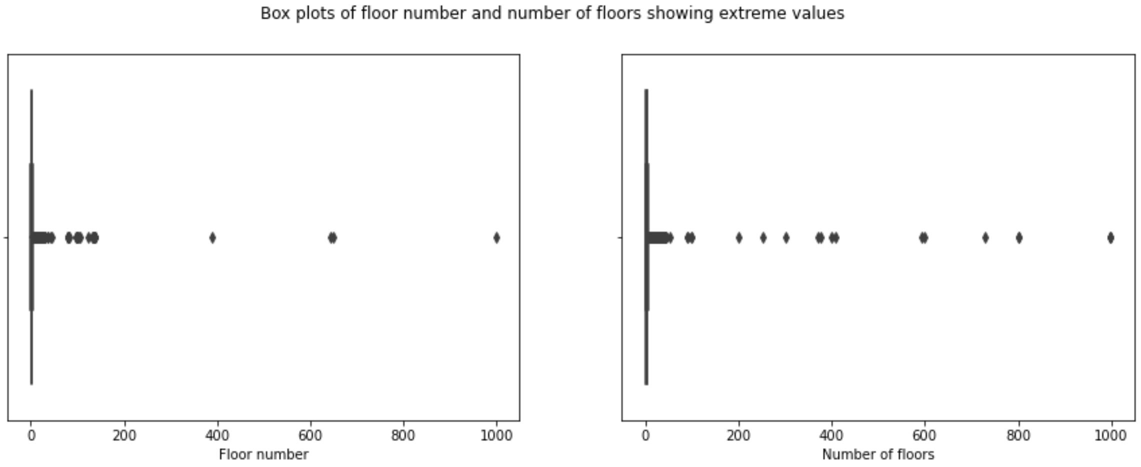


Figure 2: Boxplots of floor number and the number of floors.

In some cases, common sense relationships were used to detect outliers. For example, the minimum room size was taken to be 7.5 square meters, so any observation with a living space less than the product of this minimum room size and the number of rooms was considered to be too small to be realistic and was filtered out from the dataset.

DATA DEFINITION

For both datasets, each feature was confirmed to or transformed to its appropriate data type to allow for proper analysis. For instance, scoutID – which is the unique identification number of a listing – was given an integer data type. This feature was transformed to categorical data type because scoutID was determined to be nominal data. Other measures taken was to rename features to an appropriate name for ease of interpretation.

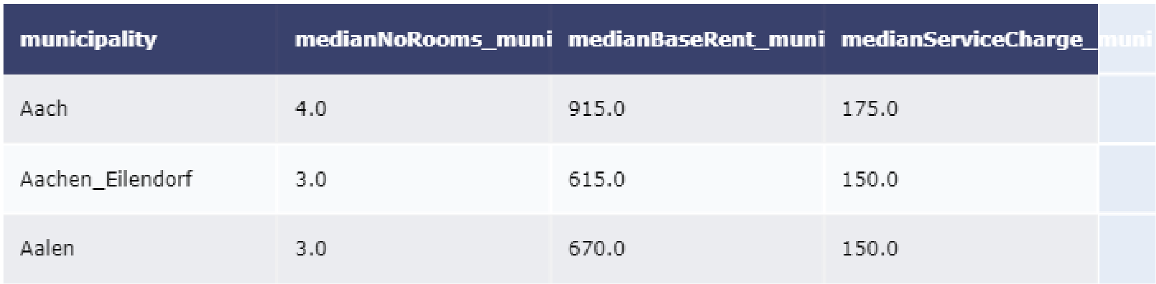
Exploratory Data Analysis

The emphasis of exploratory data analysis was to investigate the relationship between the rent and other variables in the rentals and state-features datasets in order to inform the development of a representative model that predicted rent.

Since the state-features dataset consisted of state-level data, the median rent for each state (rather than rent) was compared to these state-level features.

In addition, aggregation datasets were derived from the rentals dataset, whereby the data from numerical features were aggregated at four location levels: the state, city or town, municipality, and zip code level. Location aggregate datasets were created to understand the influence of aggregate features on the median rent at the different location levels. Median feature values and median rent was chosen as the average measure because the data distributions of the original numerical data were determined to be skewed to preclude mean aggregation. Table 1 shows a snippet of an aggregation dataset (aggregation over the municipalities) consisting of aggregate median values.

Table 1: Snippet of aggregation set showing median aggregation at the municipality level



municipality	medianNoRooms_muni	medianBaseRent_muni	medianServiceCharge_muni
Aach	4.0	915.0	175.0
Aachen_Eilendorf	3.0	615.0	150.0
Aalen	3.0	670.0	150.0

Another benefit to creating the aggregation datasets was to capture relevant location-based data before discarding some of the location features with many categories.

Four new datasets were thus created: the state-summary, city-town-summary, municipality-summary, and zip-code summary dataset.

The six datasets (rentals, state-features, state-summary, city-town-summary, municipality-summary, and zip-code summary datasets) informed our analysis of rent. It should be noted that except the rentals dataset all other datasets were explored in the context of the relationship between their features and medium rent at the respective aggregate level.

BASE RENT VS. TOTAL RENT

The rentals dataset included features referred to as 'total rent' and 'base rent'. Total rent was an aggregate feature that consisted of base rent, service charge and heating cost. This meant that total rent was highly correlated with base rent (Pearson correlation coefficient of 0.98), and only one of them would become the target variable.

It was assumed that landlords would benefit more from information pertaining to base rent rather than total rent because landlords could derive total rent from their experience with the property and its associated heating cost and service charge. Furthermore, base rent had fewer missing values than total rent, so the choice was made to have base rent as the target variable. Base rent will be referred to as 'rent' going forward.

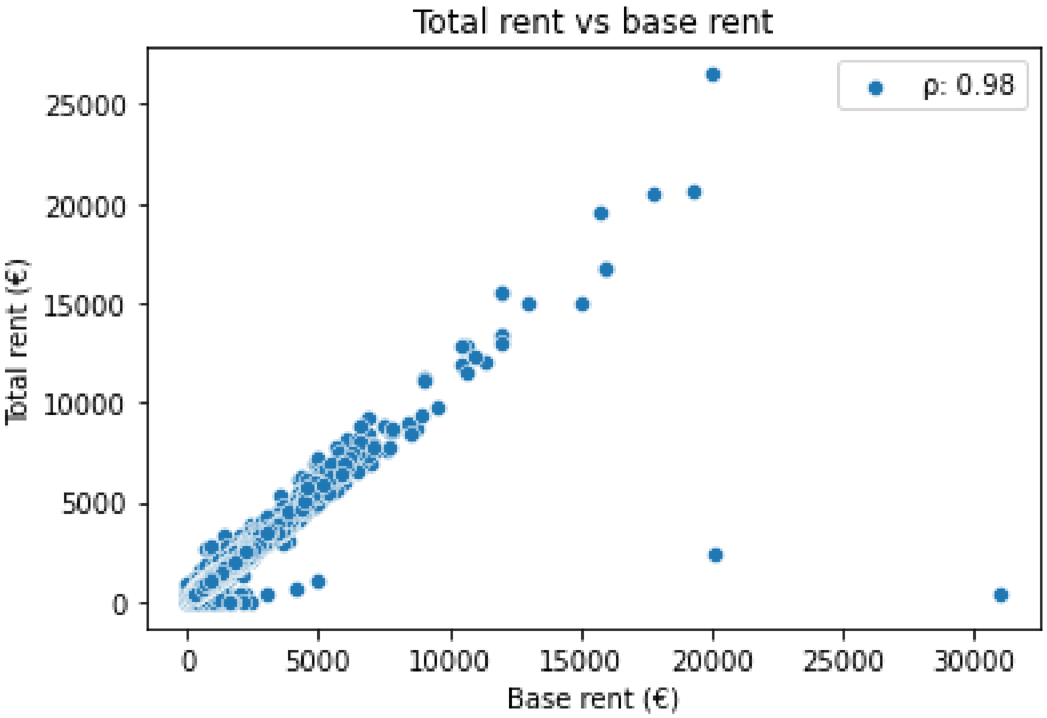


Figure 3: Scatter plot of total rent against base rent

RENT AND ITS DISTRIBUTION

The different factors affecting rent necessarily meant that one could expect a wide range of values associated with rent. The various statistics associated with rent in Germany is shown in table 2 below.

Table 2: Rent statistics

Rent statistic	Mean	Stdev	Min	25th percentile	Median	75th percentile	Max
Value (€)	647.79	515.97	1.00	338.00	490.00	799.00	39,200

The skewed nature of the rent distribution, as shown in figure 4, meant the presence of high values for rent, such that the median rent (€490) rather than mean rent (€647.79) was a better average value for all the listings.

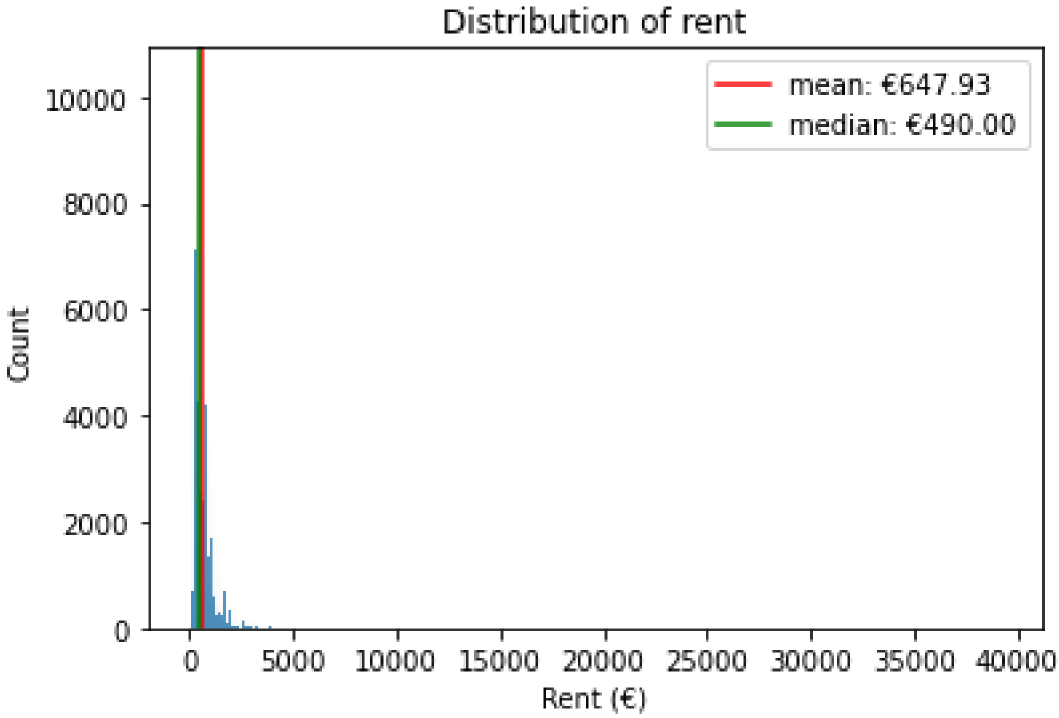


Figure 4: Distribution of rent

Furthermore, the empirical cumulative distribution function, as shown in figure 5 provided guidance to the determination of extreme rent values. Over 98% of the values for rent (corresponding to the proportion of rents less than three standard deviations from the mean rent), were values below €2,196.07.

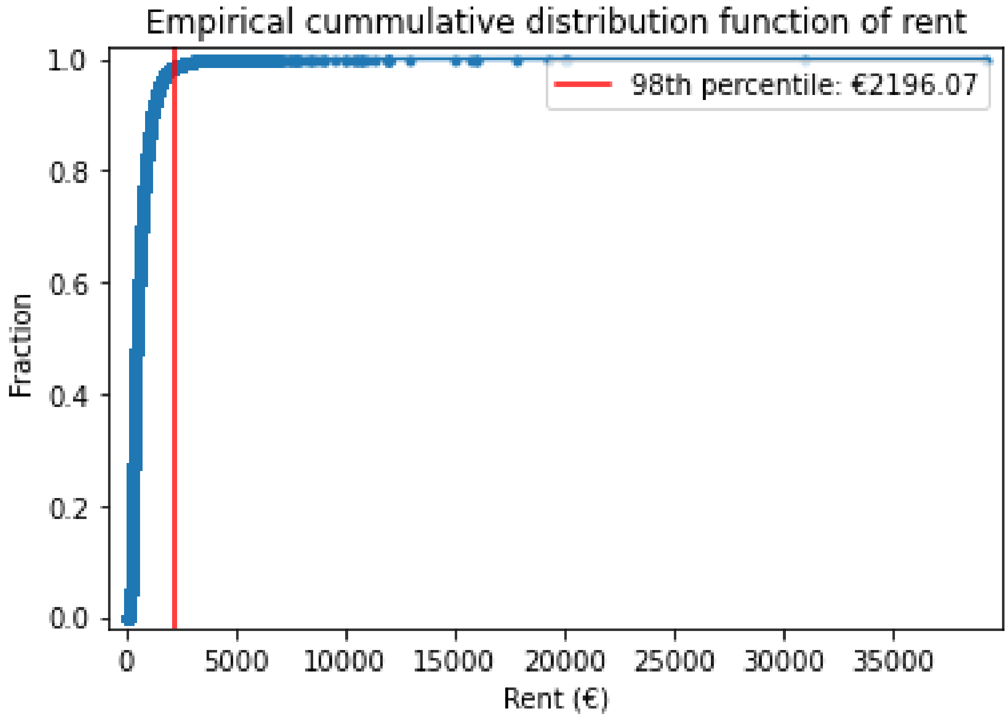


Figure 5: Empirical cumulative distribution function of rent showing rent at the 98th percentile

In sum, landlords should expect to charge substantially less rent than €2,196.07 and would charge a rent of €490 with a 50% probability. Of course, these statistics are at a country-wide level. Location variables play a key role in determining the rent that is typically charged for a property, as we will see in the following section.

RELATIONSHIP BETWEEN RENT AND TIME

The rentals dataset included three time features: the date listings were extracted from the immo data website ('date'), the year the building of the property was constructed ('yearConstructed'), and the date of the last time the property was refurbished ('lastRefurbish'). The time features all showed some patterns with rent, as will be discussed in the following section.

Relationship between median rent and 'date'

The nation-wide median rent dropped from €480.00 to €478.50 between September 2018 and May 2019 before rising to €501.00 in October 2019 and finally dropping to €495.00 in February 2020. Furthermore, these price levels did not seem to depend on the number of listings as shown in figure 6 below.

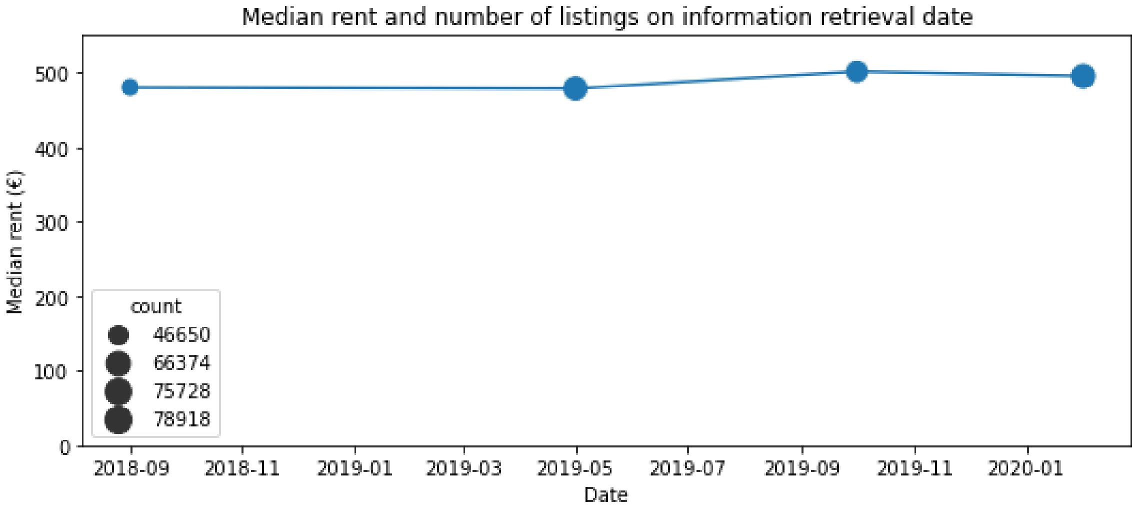


Figure 6: Median rent of samples retrieved on four dates

Relationship between rent and 'yearConstructed'

A peculiar pattern emerged when the relationship between rent and yearConstructed was reviewed. Prior to 1880, there was no pattern associated with rent and the year the building was constructed; a distinct pattern emerged post 1880, whereby rent reduced for buildings built after 1880 till the 1940s, bottoming out around 1942 and increasing thereafter to the 2020s. A steady rise in rents was particularly observed for properties built after the year 2000. Figure 7 illustrates these trends. This suggests that landlords could expect to set higher rents in proportion to the year their building was built after the year 2000.

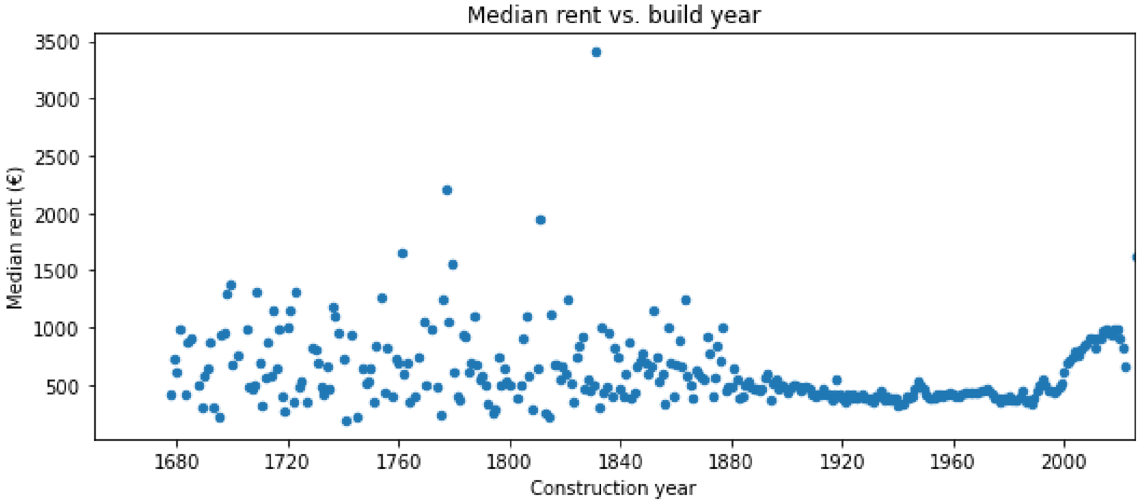


Figure 7: Median rent of buildings commissioned in a certain year

Relationship between rent and 'lastRefurbished'

Somewhat similar to the relationship between rent and yearConstructed post 2000, a rise in rents was observed for properties that were last refurbished after 1990, as shown in figure 8.

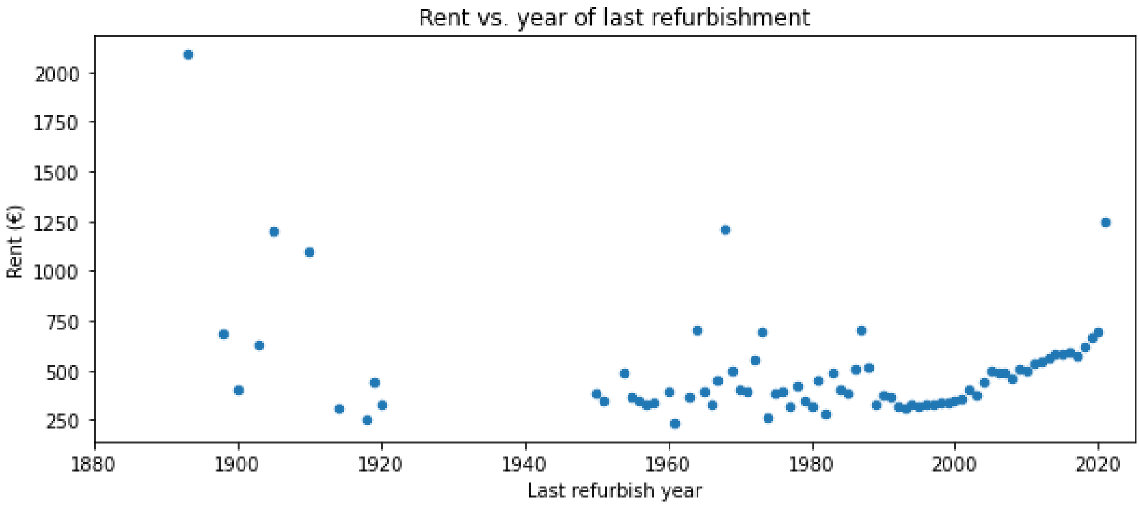


Figure 8: Median rent of apartments refurbished in a certain year

Here again, landlords could expect to set higher rent in proportion to the last time their properties were refurbished after 1990.

RELATIONSHIP BETWEEN RENT AND LOCATION FEATURES

Location! Location! Location! is a common mantra echoed in the real estate industry. Reviewing location-type features confirmed this affirmation: location seemed to play a significant factor for determining rent.

Distribution of state listings in the rentals dataset

The rentals dataset was presumed to be a robust dataset that included information from all over Germany, and this was judged to be accurate because the distribution of the number of listings per state roughly mirrored the state populations as shown in figure 9. Nordhein Westfalen had the greatest number of listings (62,740) and was found to have the highest population amongst the states (17,932,651). The states with the lowest number of listings were Bremen (2,960), Saarland (1,424), Hamburg (3,742) and Mecklenburg-Vorpommern (6,624), which also corresponded to the states with the lowest populations (i.e., 681,202, 990,509, 1,847,253, and 1,609,675 respectively). Sachsen was anomalous because it is the seventh most populous state (4,077,937) but had the second highest number of listings (57,846). This may be representative of the supply conditions in its market.

Of course, what matters most is how the number of listings and population in a state affects rent. This will be reviewed in an upcoming section.

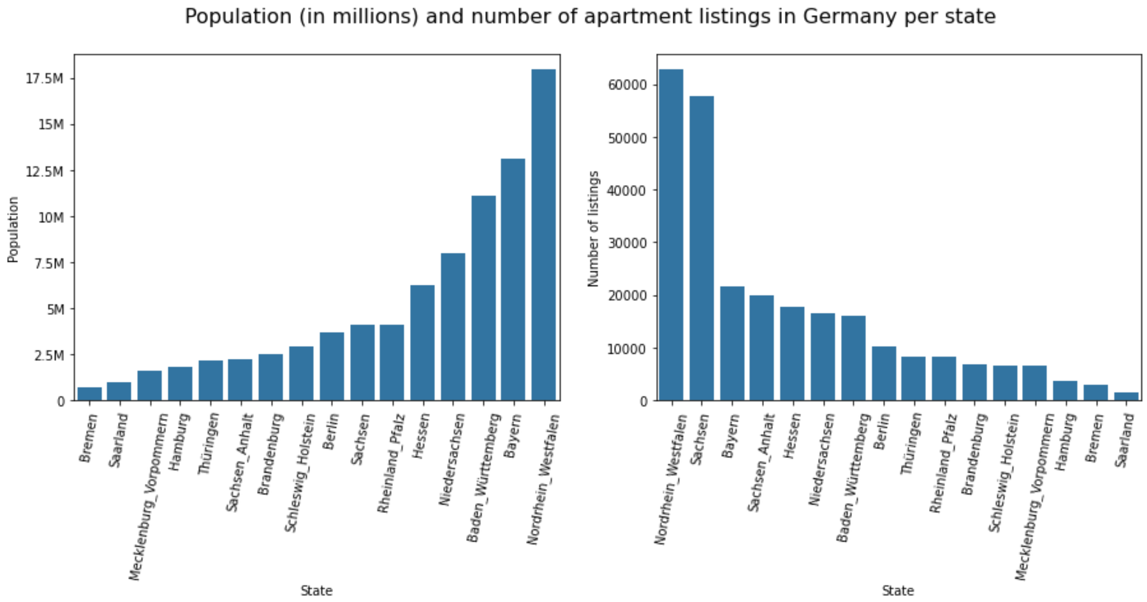


Figure 9: Bar plot of the population and number of listings for German states

Relationship between median rent and state

Five groupings of states were observed according to their median rent. These groupings are shown in table 3 below.

Table 3: Median rent range of quintile state groupings

Grouping	States	Median rent quintiles (€)	Grouping name
1	Hamburg, Berlin	850.0 to 1014.345	Very high rent
2	Baden-Württemberg, Bayern, Hessen, Rheinland_Pfalz	580.0 to 850.0	High rent
3	Niedersachsen, Saarland, Schleswig-Holstein,	500.0 to 580.0	Medium rent
4	Brandenburg, Bremen, Nordrhein-Westfalen	344.0 to 500.0	Low rent
5	Mecklenburg-Vorpommern, Sachsen, Sachsen_Anhalt, Thüringen	324.999 to 344.0	Very low rent

The boxplot further illustrates the spread of rent in the states sorted by their median rent.

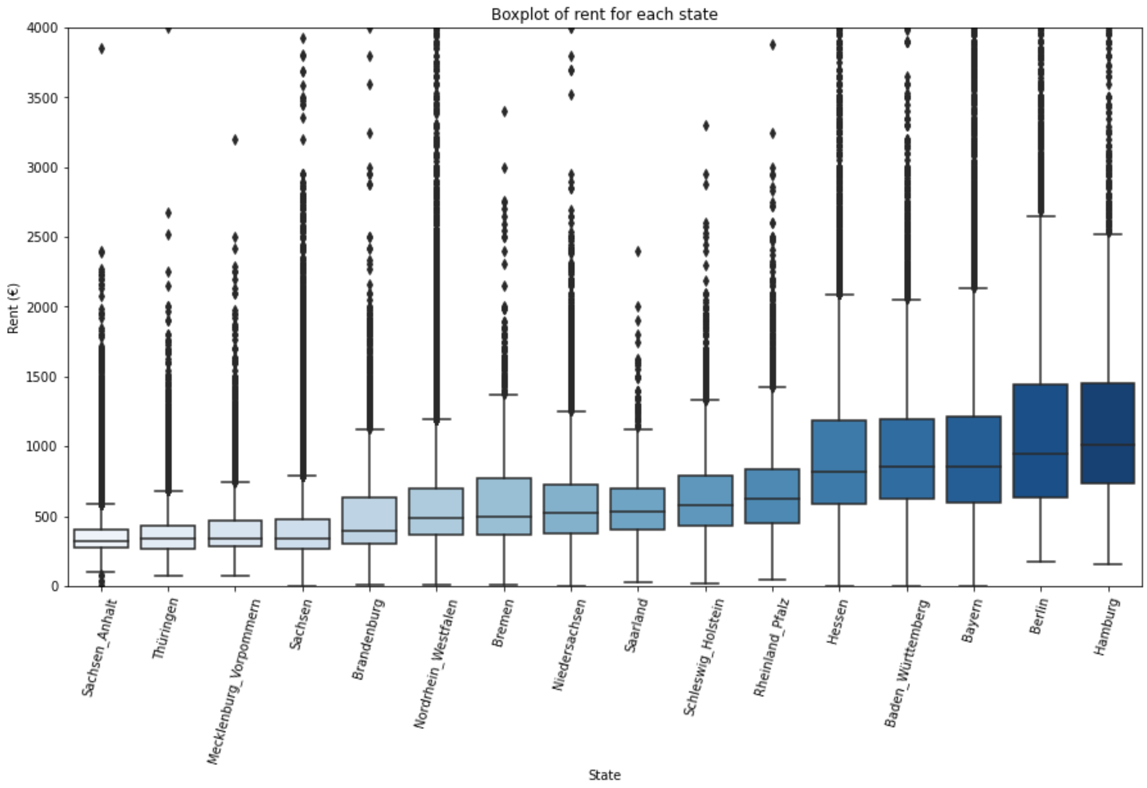


Figure 10: Box plot of rents for each state

Accordingly, landlords should expect to set higher rents in Berlin and Hamburg versus Sachsen Anhalt or Thüringen. In spite of the higher rents in the very-high rent grouping, there was much more spread of rents across their interquartile range when compared to the lower rent states.

Relationship between rent and location features at a higher spatial resolution

State-wide observations of rent were instructive, but too general to be useful to a landlord. Accordingly, the top and bottom 10 median rent locations were investigated at lower spatial levels.

Most and least expensive cities / towns to rent

München was observed to have the highest median rent and by a significant margin when compared to the next three cities. A landlord could expect to set rent around €1490 in München, which drops to €1300 in Stuttgart, Freiburg and Starnberg. The 10 most and least expensive cities / towns are shown in figure 11.

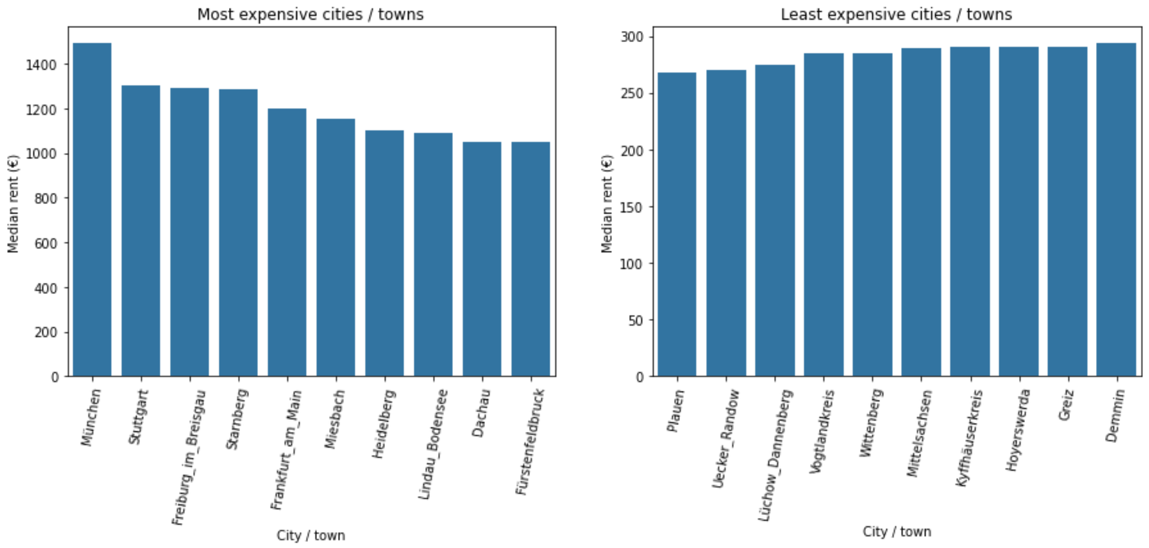


Figure 11: The ten most and least expensive cities / towns

The most expensive cities and towns were also observed to reside in the Bayern, Hessen and Baden Wurttemberg, which we considered to be high-rent states. The very high-rent states (Hamburg and Berlin) did not feature in the top 10. Except for Lüchow Dannenberg in Niedersachsen (a medium-rent state), the least expensive cities / towns resided in the very low-rent states.

Most and least expensive municipalities to rent

Even more revealing was the stark difference between the two most expensive municipalities: Gussefeld in the town of Altmark and the state of Sachsen Anhalt had a median rent of €7000, while Westheim in the town of Weißenburg and the state of Bayern had a median rent of €3000. The top and the bottom 10 municipalities are shown in figure 12.

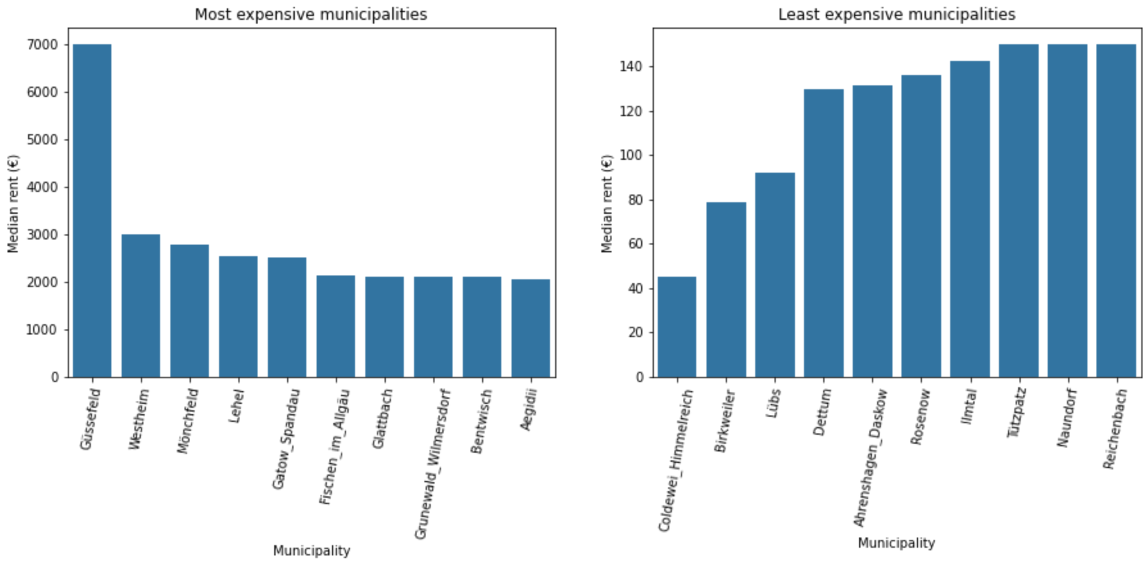


Figure 12: The ten most and least expensive municipalities

It should also be noted that Gussefeld resides within a very low-rent state. The same goes for Diedrichshagen (in Rostock, Mecklenburg Vorpommern) and Bentwisch (in Bad Doberan, Mecklenburg Vorpommern). Furthermore, one of the least expensive municipalities – Haslach – resides within a high rent-state (Kempten Allgäu, Bayern). The lack of consistency in the price levels associated with the median rent of a specific municipality and its state, suggests that rental prices are highly localized.

Most and least expensive zip codes to rent

The zip code with the highest median rent was located in Lausen Grünau in the city of Leipzig in Sachsen, with a median rent of €9000. What is interesting is that this zip code has the highest median rent but is located within a low-rent state (Sachsen). From the bar plots in figure 13, the price levels of the most and least expensive zip codes does not seem to reflect the price levels of the state, once again suggesting that rent is truly specific to the exact location of the property. That said, Hamberg and Berlin (the most expensive states) did not have zip codes in the top or bottom 10 zip codes for rent. Perhaps, this suggests that Hamberg and Berlin has lower income disparity when compared to the other states, and that individuals in these states are overall well compensated as compared to other states.

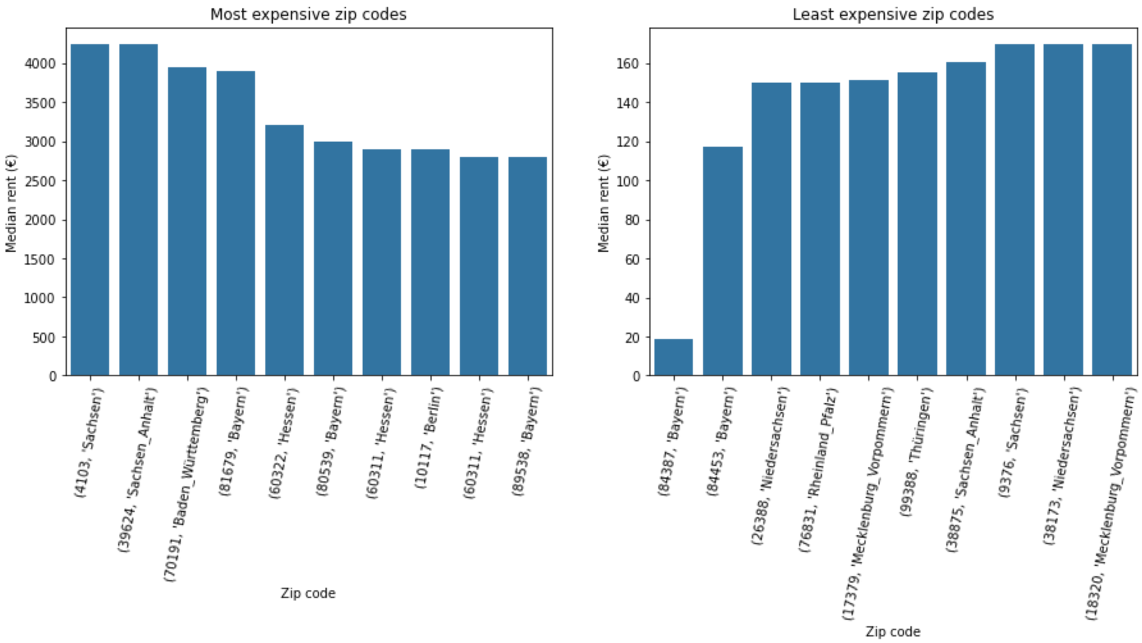


Figure 13: The ten most and least expensive zip codes and associated states

RELATIONSHIP BETWEEN RENT AND NUMERICAL FEATURES AT VARIOUS LOCATION LEVELS

Review of rent at different location levels presented some counter-intuitive results as one zoomed-in. Clearer patterns were observed at a specific location level when the rent was compared to numerical variables. Here again, rent was compared to the original numerical features at the nation-wide level, while median rent was compared to the median numerical feature at the different location levels. Only the most important relationships are discussed in the following section.

Relationship between rent and number of rooms

A positive relationship between rent and the number of rooms was observed at the nation-wide, municipality and zip code levels, but not at the state or city / town levels, as shown in figure 14 and 15. This relationship was most pronounced at the nation-wide level, with a correlation coefficient, ρ, of 0.47. The strength of the relationship, however, reduced as we zoomed-in to the lower location levels - from the municipality (ρ = 0.4) to the zip code level (ρ = 0.34).

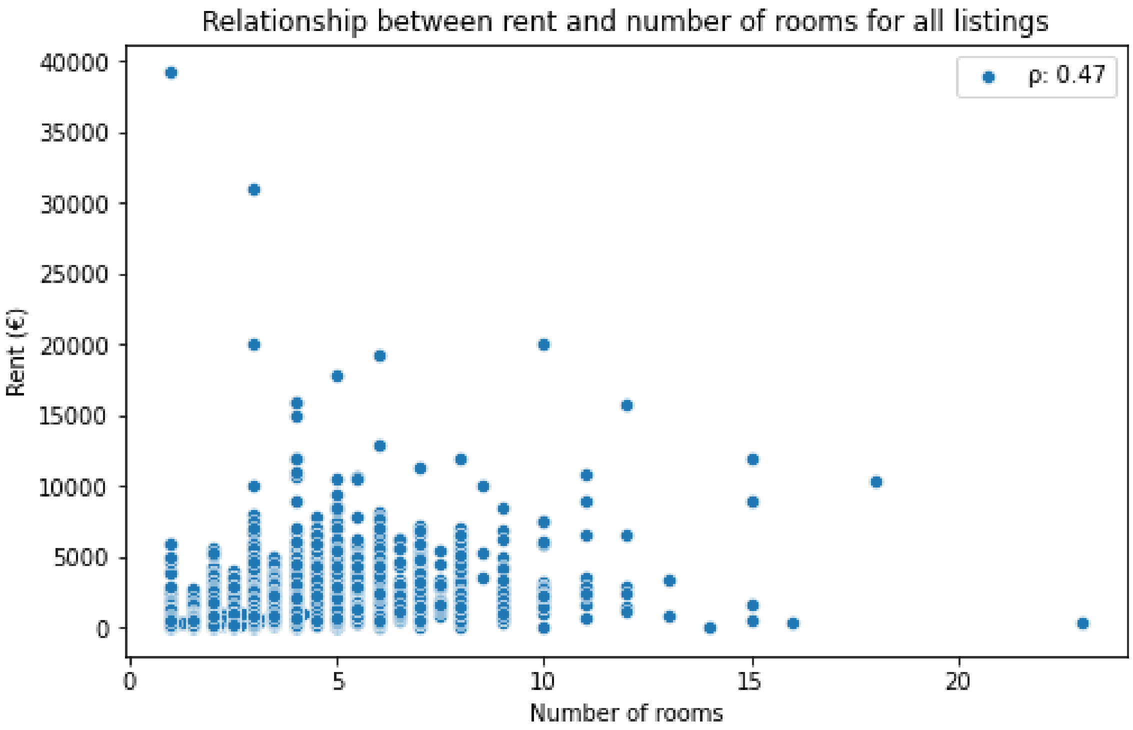


Figure 14: Scatter plot of rent and the number of rooms

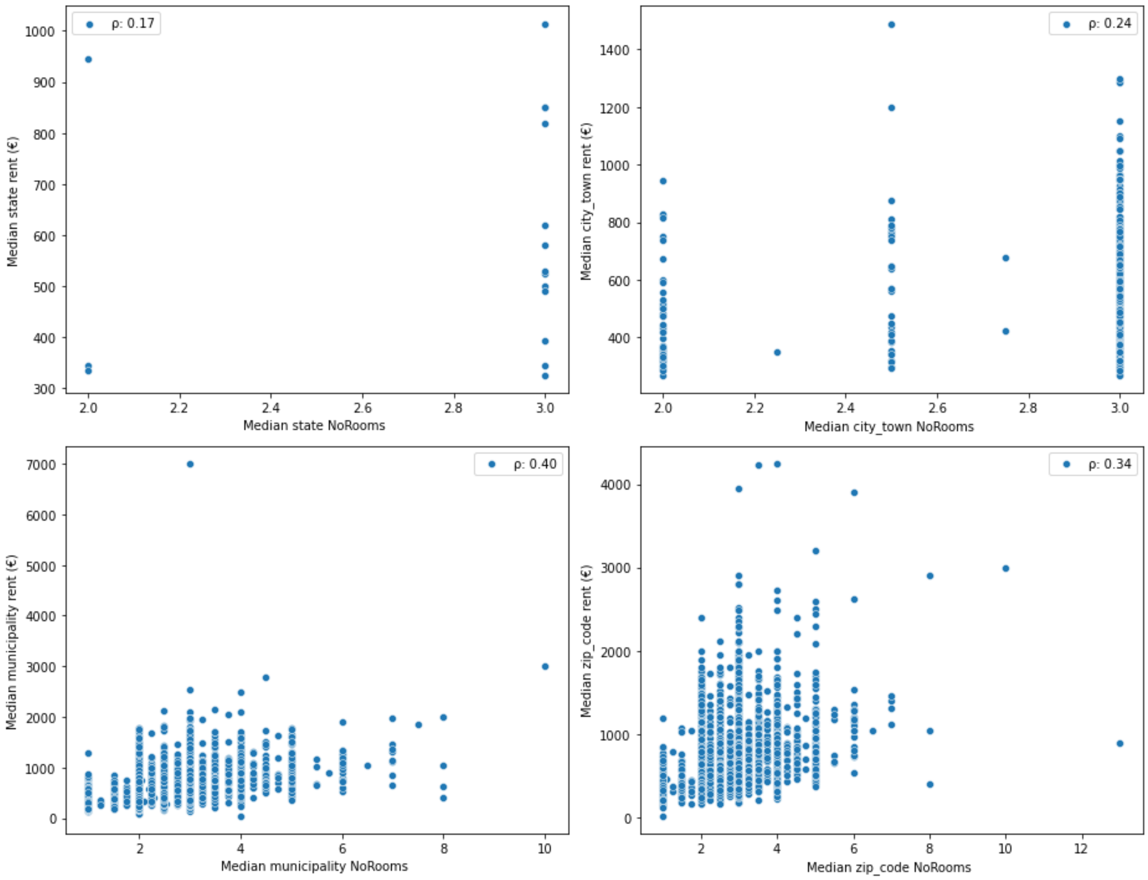


Figure 15: Scatter plots of the median rent and median number of rooms at the state, city / town, municipality and zip code level

Relationship between rent and living space

A strong relationship was observed between rent and the living space when all the listings were considered, with a correlation coefficient of 0.72 as shown in figure 16. A strong correlation was also observed between the living area and rent at the aggregate location levels as shown in figure 17, even though there was a reduction in the strength of this relationship as we went down these levels. The strongest relationship was observed at the state aggregate level, with a correlation coefficient of 0.83.

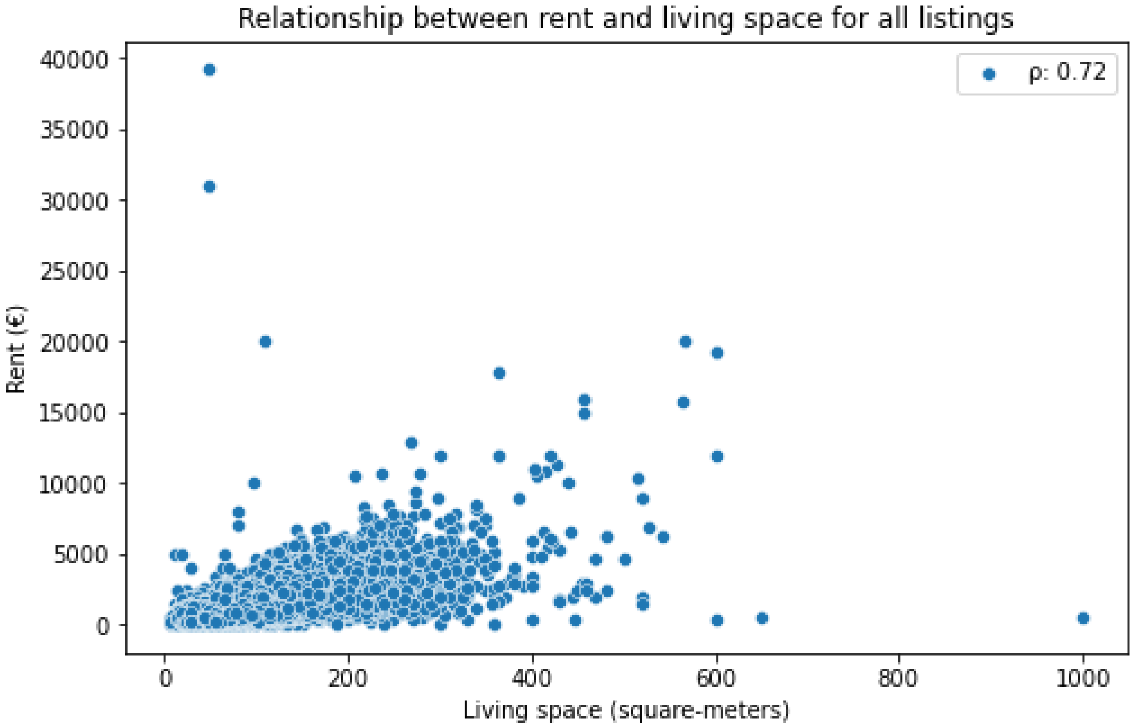


Figure 16: Scatter plot of rent and living space

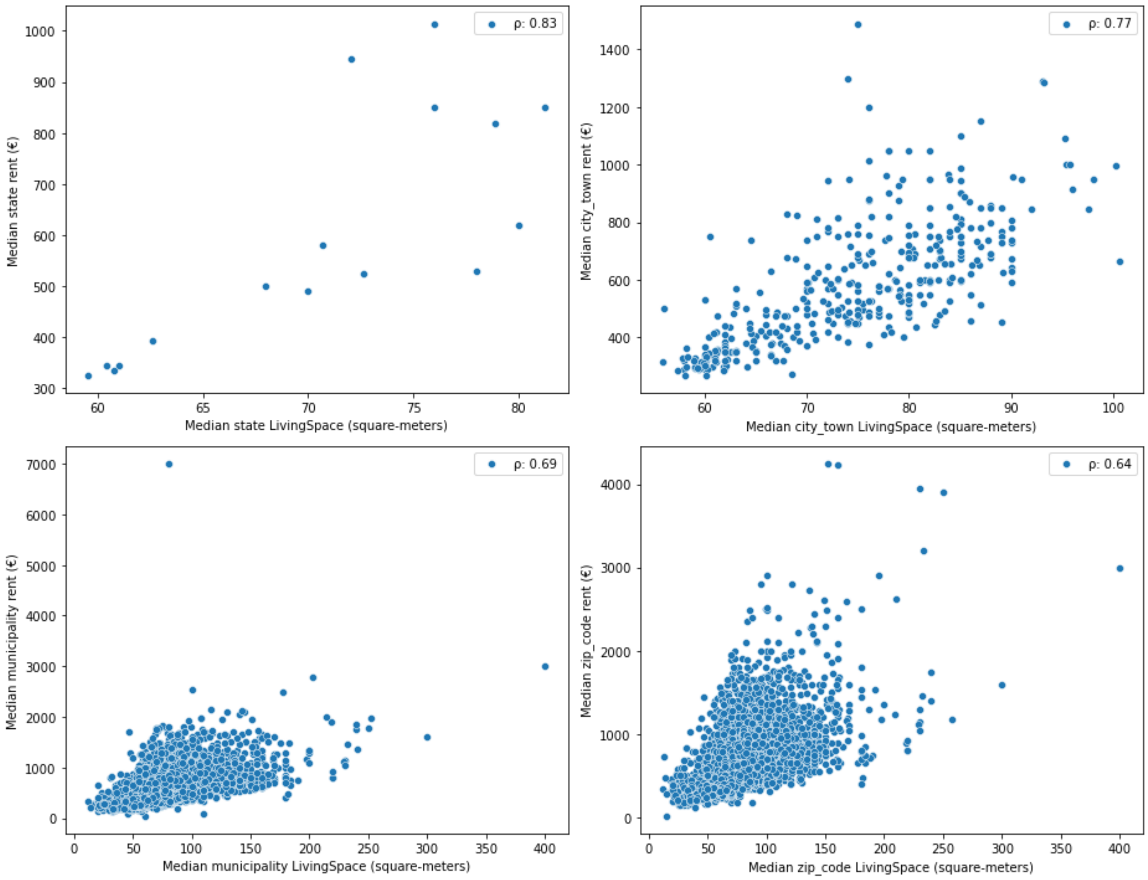


Figure 17: Scatter plots of the median rent and median living space at the state, city / town, municipality and zip code level

Relationship between rent and service charge

A significant positive relationship was observed between the rent and service charge for all listings, with a correlation coefficient of 0.65, which corresponded to correlation coefficient between the median rent and median service charge at the zip code level, as shown in figure 18 and 19 respectively. There was, however, a very strong relationship between the median rent and median service chart at the state level, with a correlation coefficient of 0.91, the highest observed in the analysis.

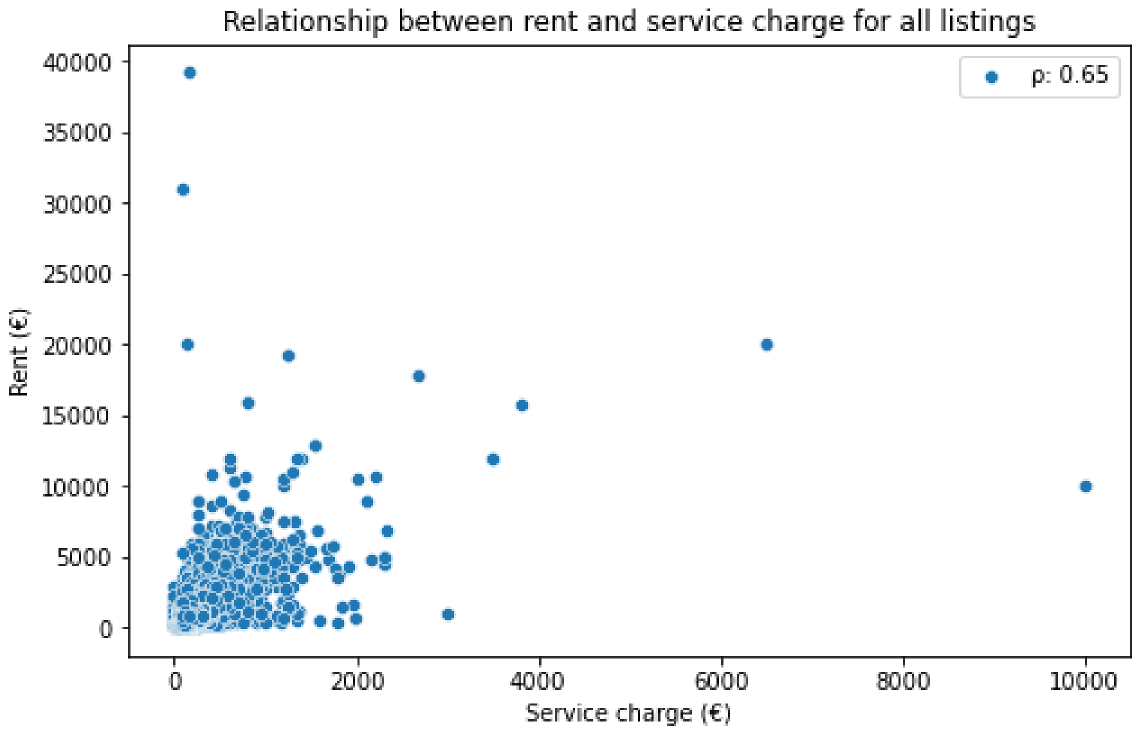


Figure 18: Scatter plot of rent and service charge

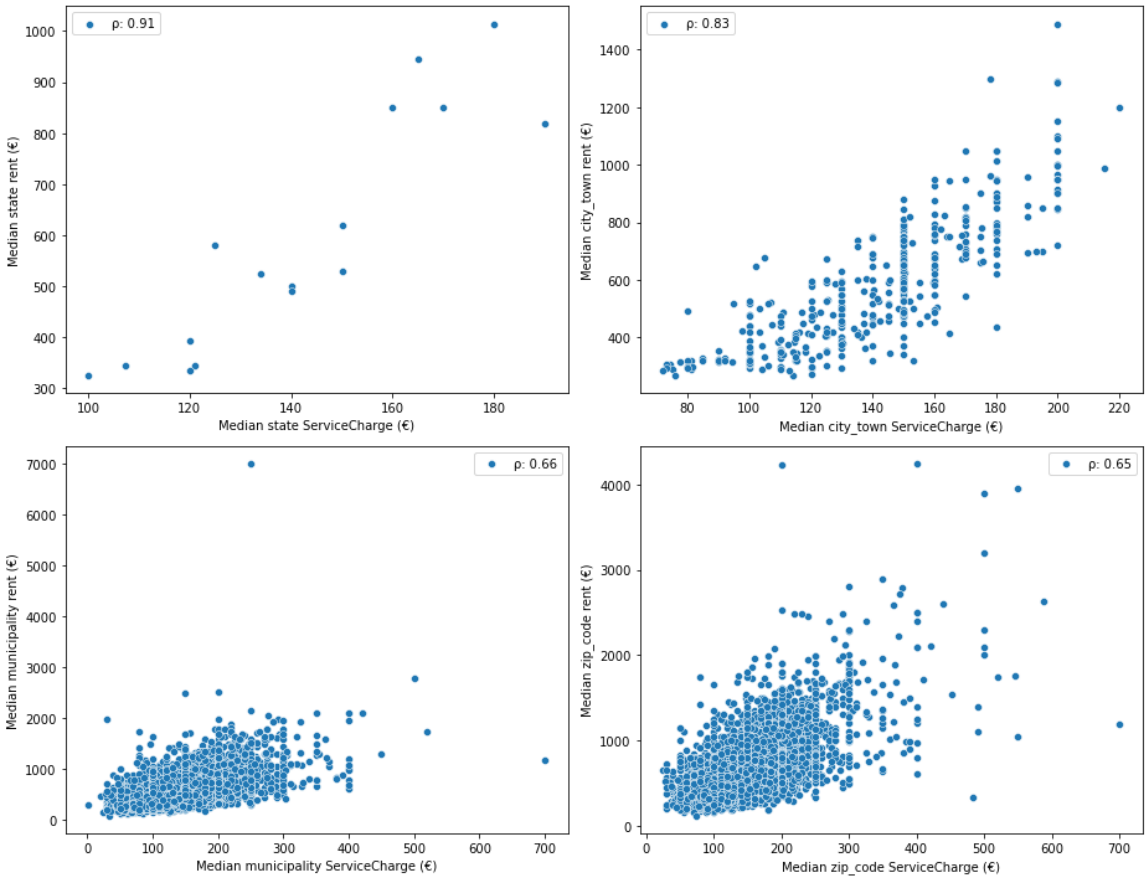


Figure 19: Scatter plots of the median rent and median service charge at the state, city / town, municipality and zip code level

Relationship between rent and heating costs

The only relationship observed between heating costs and the rent was seen at the state aggregate level, with a correlation coefficient of 0.82 between the median heating cost and median rent, as shown in figure 20. The state-wide relationship is likely due to the fact that weather patterns are more distinct at a state-wide level.

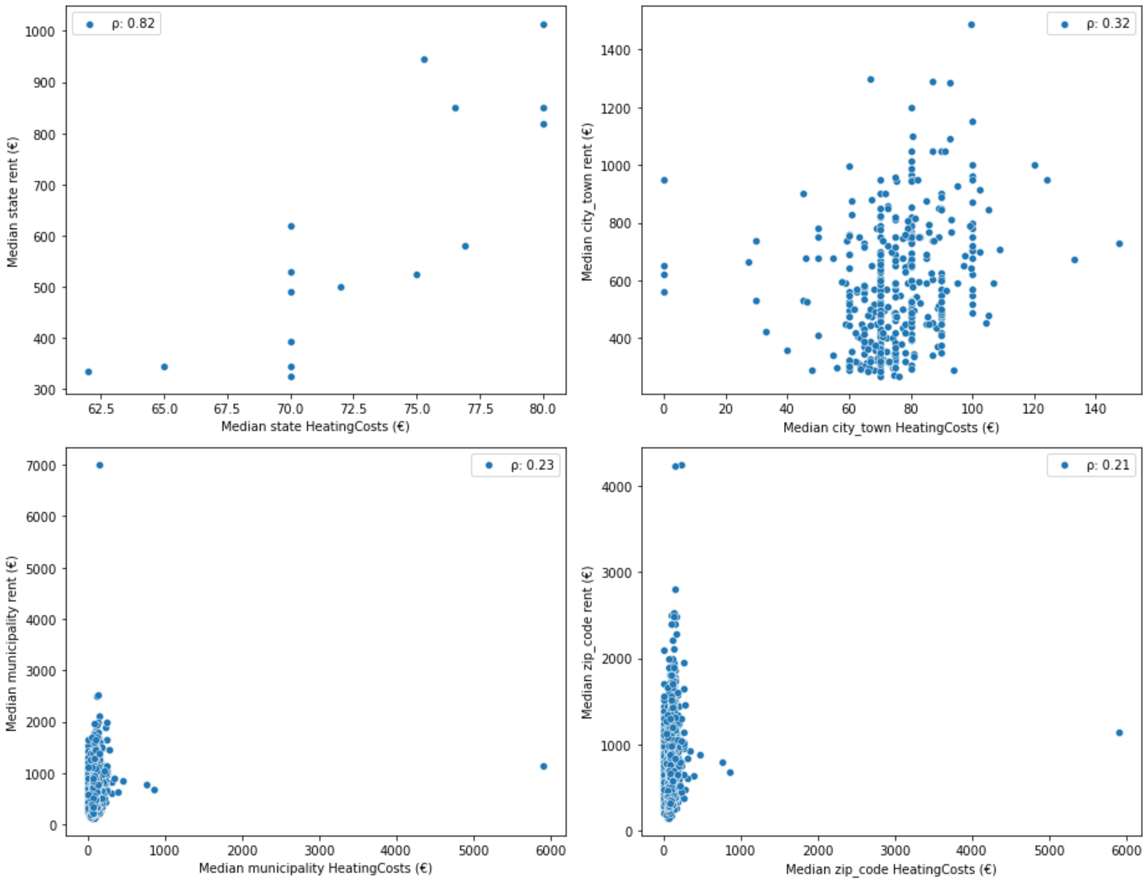


Figure 20: Scatter plots of the median rent and median heating costs at the state, city / town, municipality and zip code level

Relationship between rent and thermal characteristics

The thermal characteristic feature defines the energy demand of an apartment and is thus an indicator of the energy efficiency of the apartment. There was not much of a relationship observed between the rent and the thermal characteristic feature at the nation-wide, state, municipality or zip code levels; however, a negative relationship was observed between the median rent and median thermal characteristic at the city / municipality level, with a correlation coefficient of -0.32, suggesting that the rent decreased with increasing energy demand, as shown in figure 21.

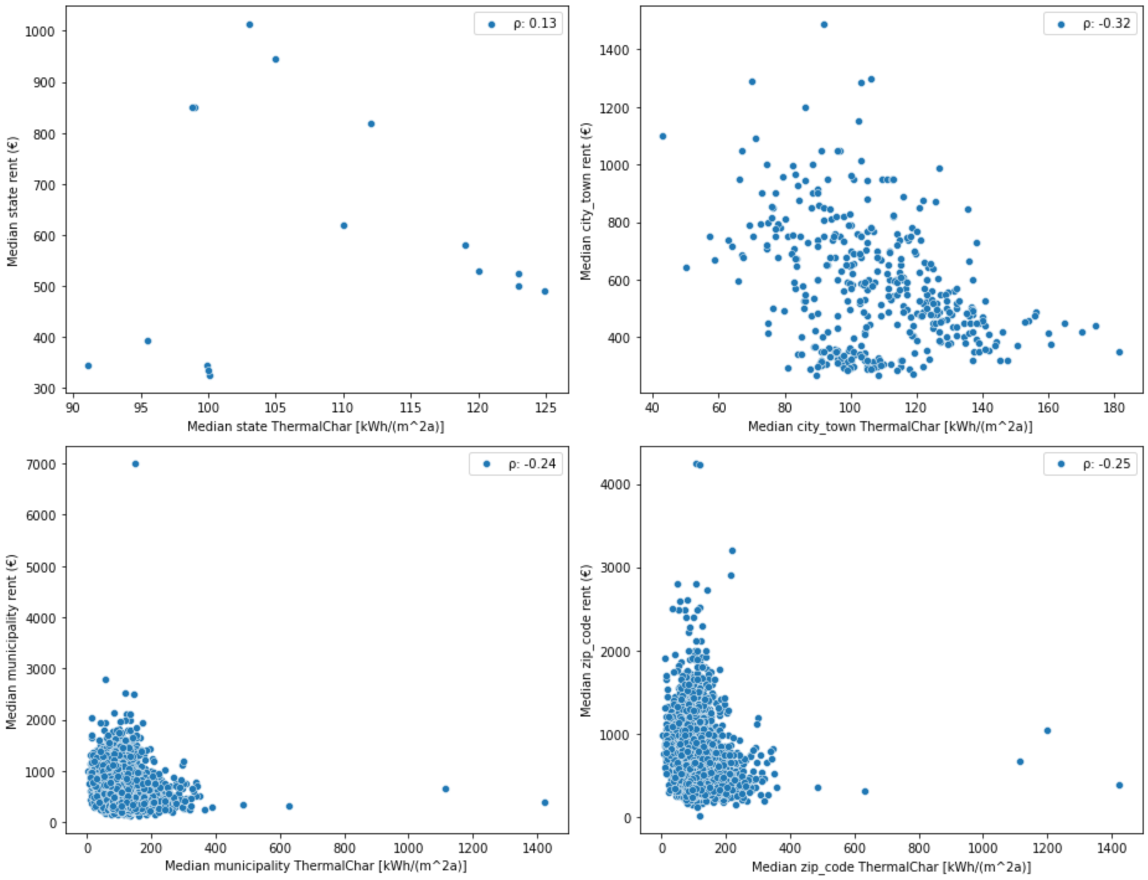


Figure 21: Scatter plots of the median rent and median thermal characteristic at the state, city / town, municipality and zip code level

Relationship between rent and the number of pictures

It was interesting to observe a relationship between the number of pictures provided for a listing and the rent. This was not obvious at the nation-wide level; however, a slightly positive trend was seen between the median rent and median number of pictures at the state, city / town, municipality, and zip code level, with correlation coefficients of 0.73, 0.58, 0.33 and 0.34 respectively, as shown in figure 22.

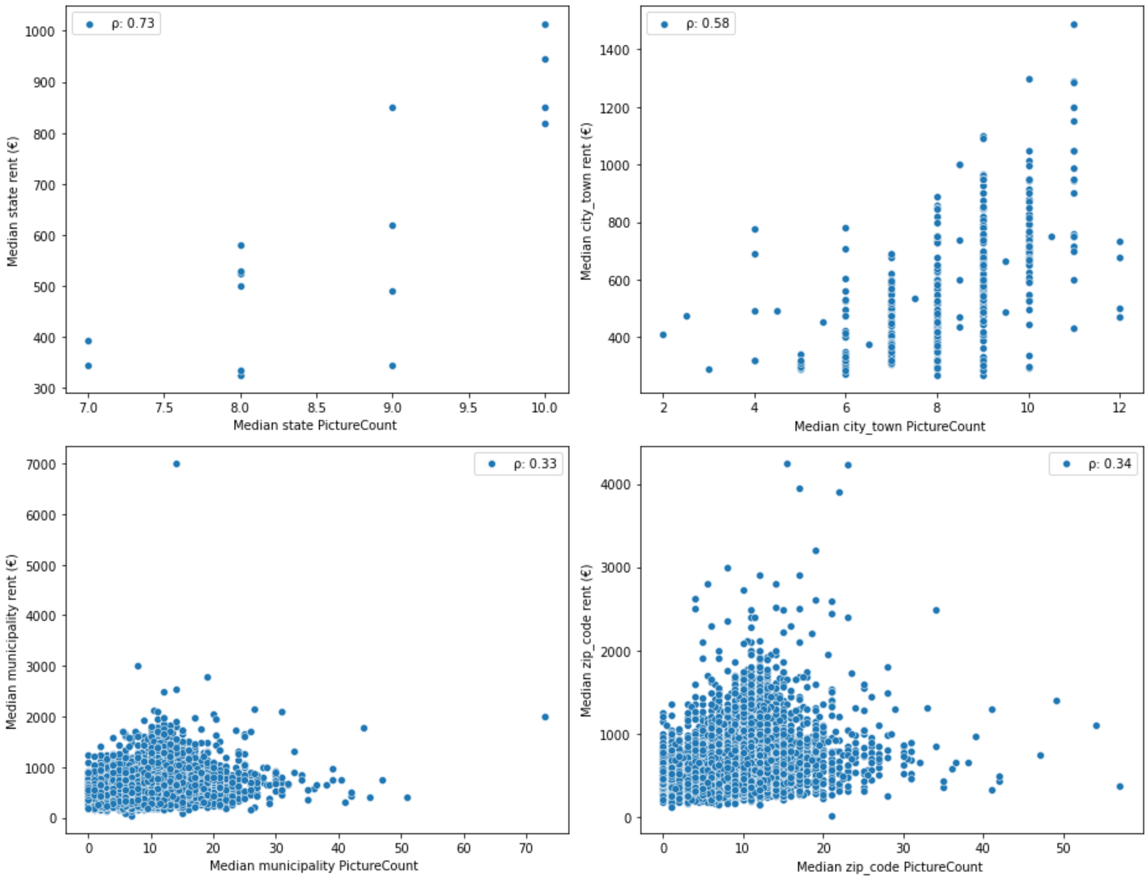


Figure 22: Scatter plots of the median rent and median picture counts at the state, city / town, municipality and zip code level

VARIATION OF RENT WITH CATEGORICAL FEATURES

Boxplots of rent stratified by categorical values were used to understand the influence of categorical features on rent. The interior quality, condition and type of apartment showed the most distinct relationships to rent.

Rent vs. interior quality

Interior quality is an indication of the degree of elegance of a property, the lowest level being ‘simple’ and the highest level being ‘luxury’. The boxplots of rent for different interior qualities suggested higher rents set for properties with either a ‘luxury’, or ‘sophisticated’ interior quality, as shown in figure 23.

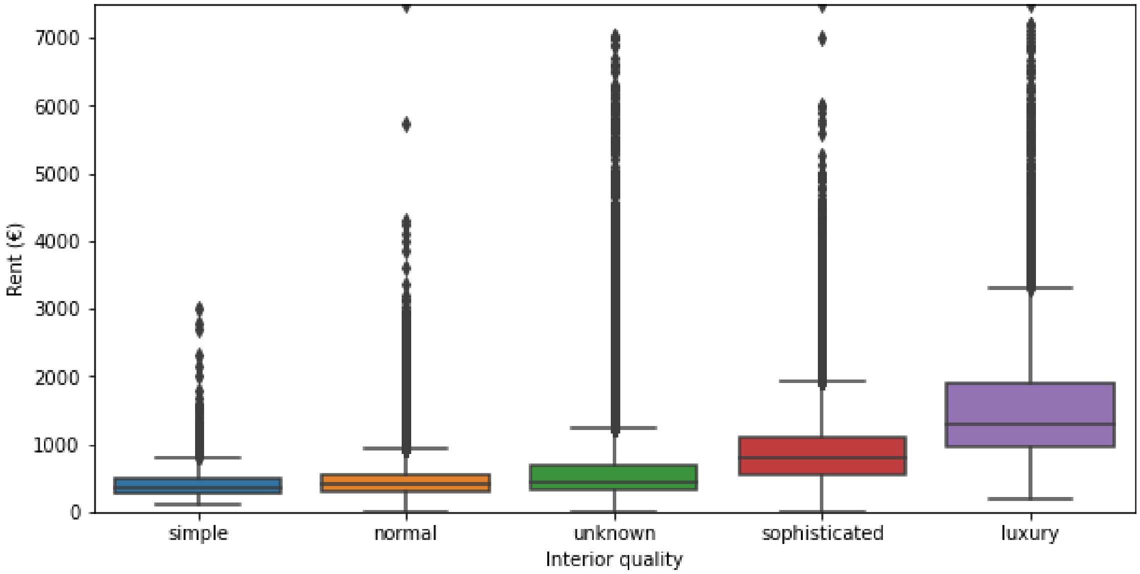


Figure 23: Box plots of rents per interior quality of apartment

Rent vs. condition

Condition indicates the quality of the property and ranges from first-time-use to ripe-for-demolition. Boxplots of rent for different conditions showed that higher rents were demanded for first-time-use and mint condition properties, as shown in figure 24.

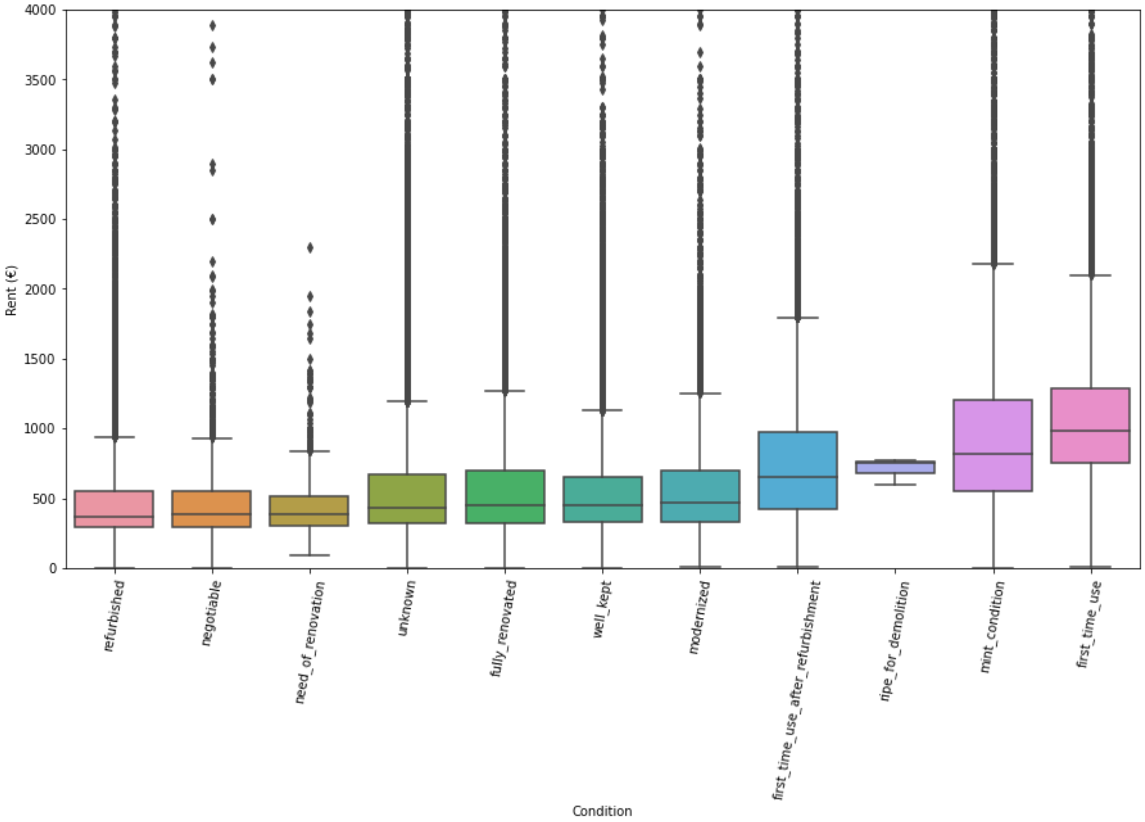


Figure 24: Box plot of rents per condition of apartment

Rent vs. the type of flat

Type of flat indicates the architectural design of the property. It included such values as ‘apartment’, ‘loft’, and ‘penthouse’. The boxplot of rents for different flat styles showed the penthouse and lofts commanding higher rents than the other flat styles, as shown in figure 25.

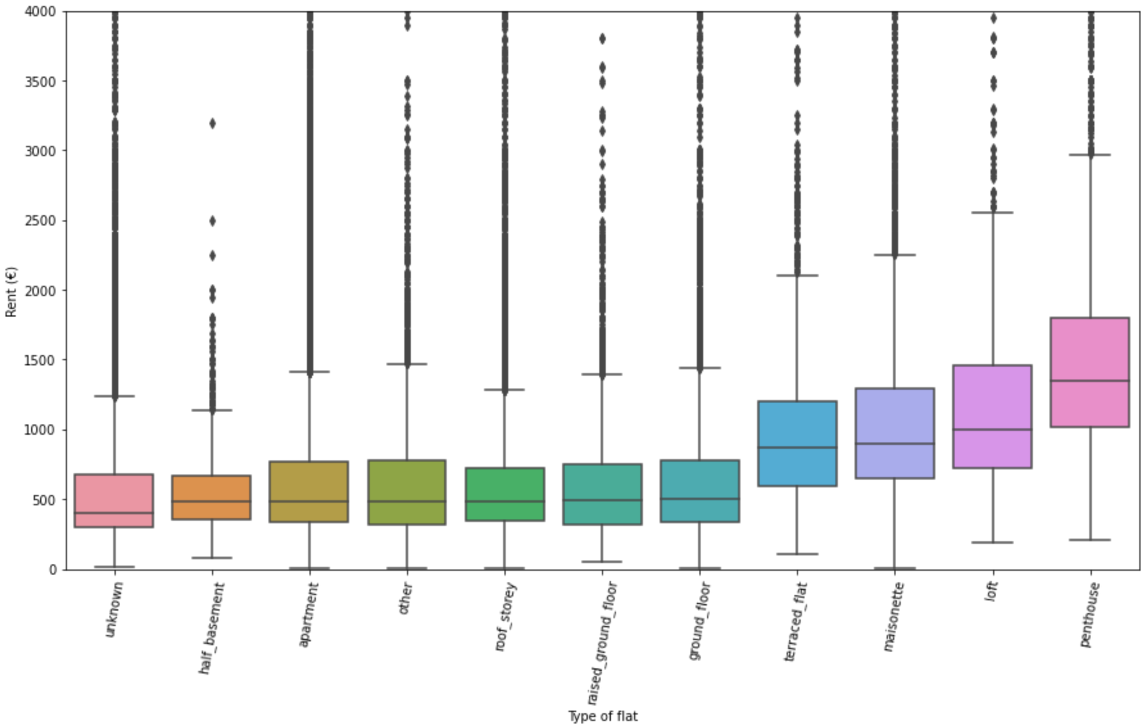


Figure 25: Box plots of rents per type of flat

REVIEW OF RENT AND FEATURES IN THE STATE-FEATURES DATASET

The state-features dataset provided geo-economic information at the state level that was valuable in our rent analysis. The most revealing features were the population per state area, GDP per capital, HDI (Human Development Index), total number of listings per state population, total living area per state population and the total number of rooms per state population. These features will be examined in the following section.

Median rent vs. GDP per capital and the HDI (Human Development Index)

The GDP per capita represents the economic strength of the state, while the HDI reflect the standard of living for the state. Both these features had a positive relationship with rent depending on the state, implying that states with a higher GDP per capita and HDI had higher median rents, as shown in figure 26.

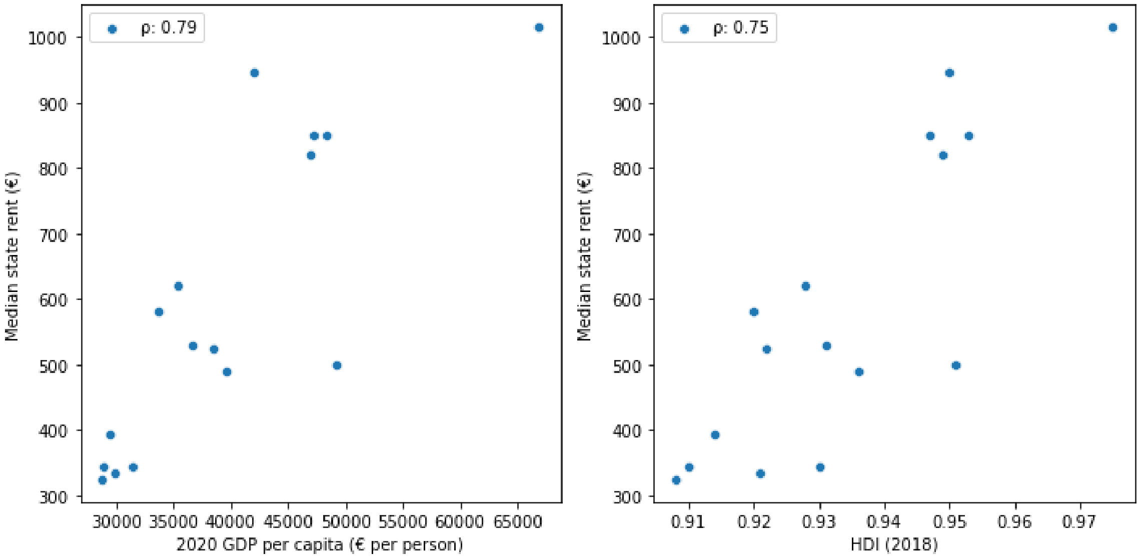


Figure 26: Scatter plots of median state rent vs. GDP per capita and HDI

Median rent vs. population per state area

No clear relationships were observed between the median rent of a state and either its population or its area; however, a distinct relationship was observed between the median rent and the population per area (population density), where the higher the population density of the state, the higher the median rent. This relationship was not linear however, but a logarithmic growth curve, wherein the median rent tapered with increasing population density after a rapid initial incline. This is shown in figure 27.

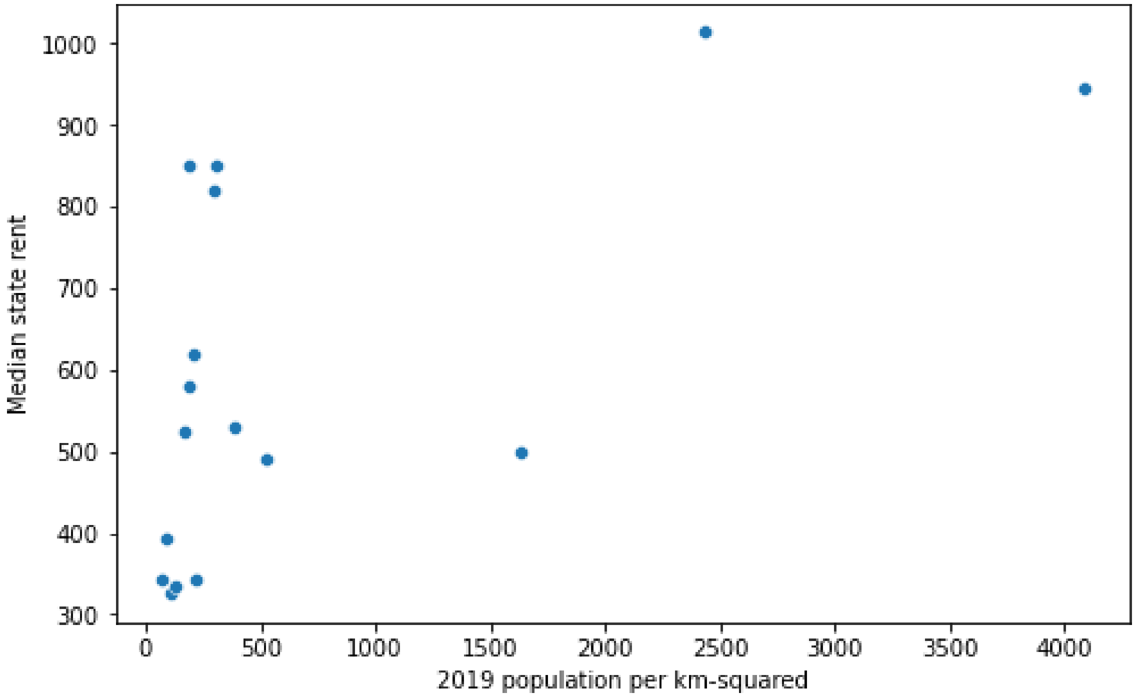


Figure 27: Scatter plot of median state rent and population per square kilometers

Median rent vs. total number of listings, total living area, and total number of rooms per state population

A distinct exponential decay curve was observed between the median rent and the total number of listings per state population, the total living area per state population, and the total number of rooms per state population as shown in figure 28 and 29. This suggest that as the number of these aggregate values related to quantities supplied per population increase there is a rapid decrease in the median rent supported by the state.

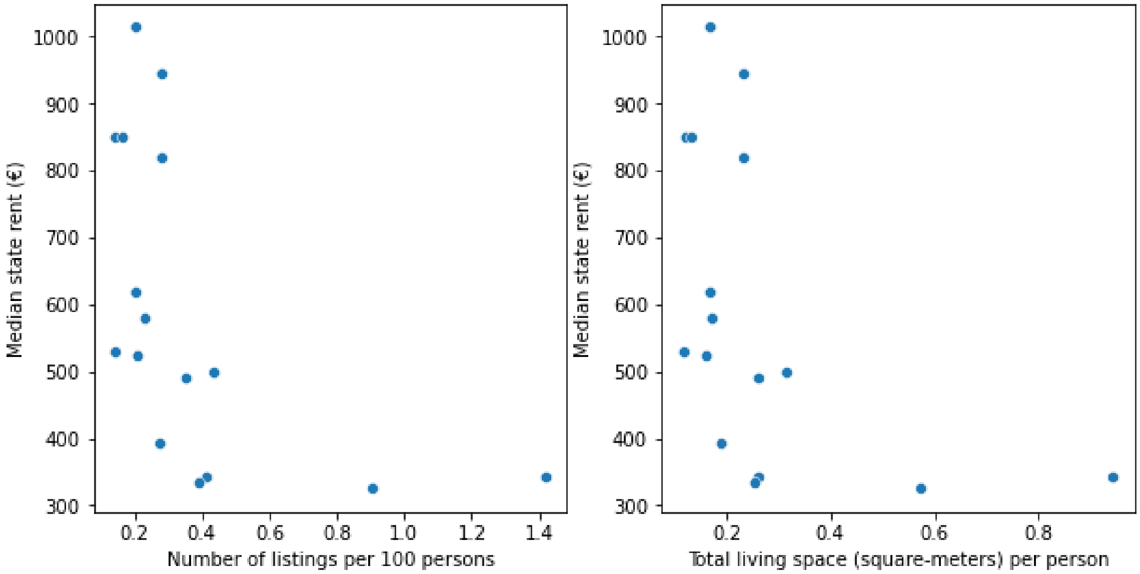


Figure 28: Scatter plot of median state rent vs. number of listings per 100 persons and total living space per person

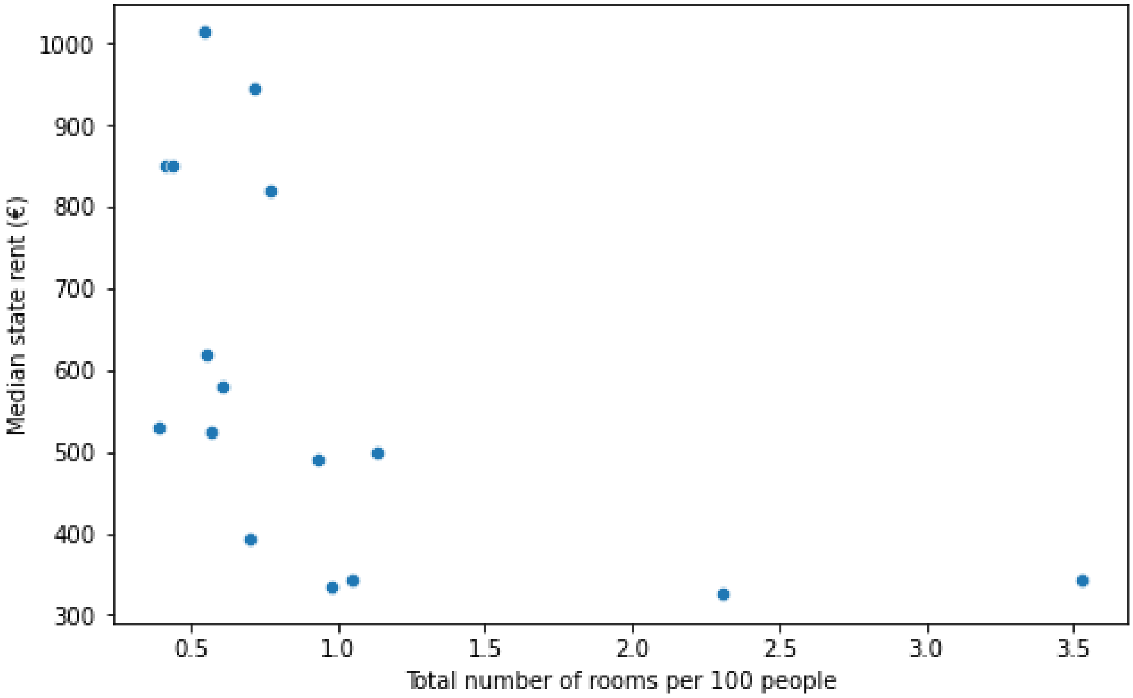


Figure 29: Scatter plot of median state rent and total number of rooms per 100 people

Exploratory data analysis revealed features that would likely be relevant to a rent prediction model. All datasets were merged into one for further processing.

Data pre-processing and training data development

DATA PRE-PROCESSING

Pre-processing was fairly straightforward: the numerical features were scaled after missing values were imputed with median values. The missing values of the categorical features were already coded with an 'unknown' identifier during wrangling, so the only step required for them was dummy encoding.

TRAINING DATA DEVELOPMENT

Application set

A new dataset was constructed from five random samples selected from the pre-processed dataset and was called the application-set. The intent of the application set was to visually inspect the results of the eventual model because the application-set mirrored the type of information a typical landlord would have. The eventual model was then used to determine predicted rent values that were compared to the actual rent values. This will be discussed in an upcoming section.

Training and test set

The remaining data was divided into a training and test set at a 70% / 30% ratio. The training set was used to create a model, while the test set was used to assess the model's performance. Each set was further divided into a dataframe of features and an array of the target feature (rent), such that there were four datasets: X_train and y_train representing the features and rents for the training set; and X_test and y_test representing the features and rents of the test set.

METRICS AND MODEL ASSESSMENT STRATEGY

Metrics

The r-squared (r2) score and the mean absolute error (mae) score were chosen as the preferred assessment metrics. The r2 score of a model provided an understanding of the proportion of variation in the rent values attributable to the model. The mae score of a model provided an understanding of the magnitude of errors that resulted from applying our model. The mae score was chosen rather than the root-mean-square-error score because the weight associated with a large error in rent is the same as the weight associated with a small error in rent. Together, these two metrics were incorporated into the model assessment strategy.

Model assessment strategy

A 5-fold cross-validation approach was used to simultaneously train and test folds of the training set to yield 5 cross-validation scores. A mean and standard deviation was computed from the array of scores to provide a sense for the typical performance of the model and the bounds of this performance. Both the mae and r2 scores were computed for each model undergoing the cross-validation evaluation. The r2 scores were used to review the different modalities of a particular algorithm, while the r2 and mae scores were used for final model selection. The choice of five folds was a judgement call based on the size of the dataset, computational capabilities of the processor and the belief that the scores from five folds of the training dataset were sufficient to reveal overfitting concerns.

The final assessment of a model was conducted with the test set, where the final metrics provided an indication of the model's performance on unseen data.

BASELINE MODEL CREATION AND ASSESSMENT

A linear regression algorithm was used to create a baseline model after the pre-processing and training data development steps. The model was trained on the training set and assessed with the test set. No cross-validation scores were computed for the model. The overall model performance on the training set and test set is shown in table 4.

Table 4: Baseline model performance scores

	R2 score	Mae (€)
Train set	0.73	146.36
Test set	0.75	147.29

The baseline test scores of 0.75 and €147.29 suggested that there was a relationship between the features and rent, such that rent could be modeled.

EXTENDED MODELLING PLAN

With a baseline test r2 score of 0.75 and mae score of €147.29, there were opportunities to improve performance. The steps that were taken to improve performance included:

- Feature selection
- Trialing other algorithms
- Tuning model hyperparameters.

Model optimization, selection and application

MODEL OPTIMIZATION

Knowing that the test scores to beat were an r2 score of 0.75 and a mae of €147.29, two linear algorithms (lasso regression and ridge regression) and two tree-induction algorithms (random forest and XGBoost) were trialed. The lasso and ridge regression algorithm were chosen as an extension of the linear regression model because they inherently penalized useless features for improved performance. The random forest and

XGBoost models required feature selection to optimize performance. Hyperparameter tuning was also used to further improve the performance of the tree-induction models.

Linear algorithms

Lasso algorithm

The model derived from the lasso algorithm produced a cross-validated r2 score of 0.73 ± 0.05 on the training set and a test score of 0.75 (the same score as the baseline model). The feature coefficients are shown figure 30.

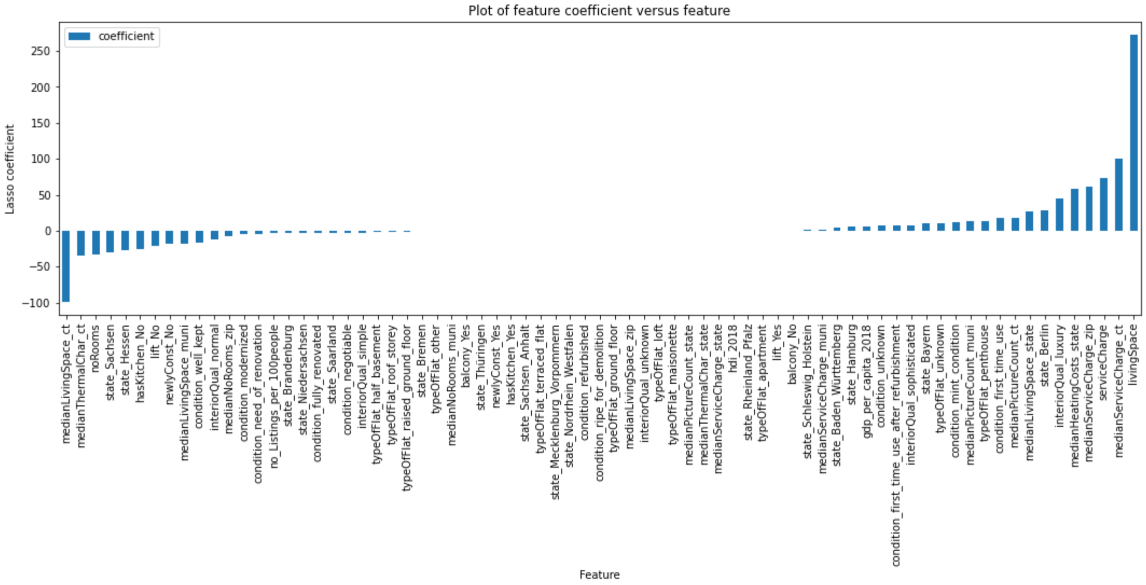


Figure 30: Features and their coefficient for the lasso model

The top 10 features from the lasso model were – the living space, median city service charge, median city living space, service charge, median zip code service charge, median state heating cost, luxurious interior quality, median city thermal characteristic, number of rooms and the state of Sachsen.

Ridge algorithm

The model trained with the ridge algorithm produced the same exact r2 scores as the lasso model - cross-validated r2 score of 0.73 ± 0.05 on the training set and a test score of 0.75. The feature coefficients are shown figure 31.

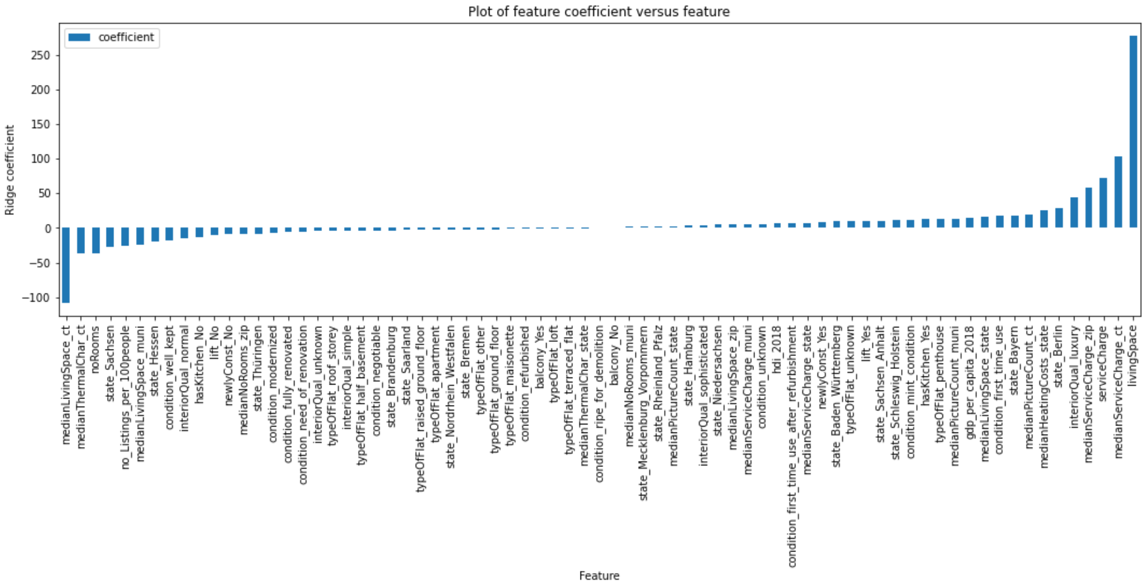


Figure 31: Features and their coefficient for the ridge model

The top 10 features from the ridge model were – the living space, median city living space, median city service charge, service charge, median zip code service charge, luxurious interior quality, median city thermal characteristic, number of rooms, the state of Berlin, and the state of Sachsen.

Summary of findings from linear models

The ridge and lasso models did not improve the performance on the baseline linear model. This suggests that a linear model could only take us so far in modeling the data. Nonetheless, there was consistency in the top 10 features observed for the lasso and ridge models. The living space and service charge emerged as the most important features of the dataset.

Tree-induction algorithms

Tree-induction algorithms followed a different approach than the linear algorithms in terms of model development. The random forest and XGBoost algorithms were used to train an initial model, whose coefficients were used to select the most important features based on a threshold value. The pre-selected features were then used to train and tune the hyperparameters of the random forest and XGBoost algorithms for the creation of a final random forest and XGBoost model. Model assessment was conducted throughout the aforementioned stages.

Random forest algorithm

Initial random forest model and feature selection

The initial random forest model yielded a cross-validation r2 score of 0.81 ± 0.05 on the training set and an r2 score of 0.84 on the test set. A coefficient value of 0.001 was chosen to pre-select the most important features. With feature selection, the test performance slightly improved, with an r2 score of 0.85.

Hyperparameter tuning with feature selection

A new model was developed from the pre-selected features and the results of tuned parameters. The number of trees and maximum depth of each tree were the parameters that were optimized using a random search algorithm on 5-fold cross validation set, with the r-squared score as the assessment metric. The optimal parameters from the search grid were 20 trees and a maximum tree depth of 20.

The eventual model yielded a slightly better cross-validation r2 score of 0.82 ± 0.05 on the training set and an r2 score of 0.86 on the test set.

The feature importance of the final model is shown in figure 32.

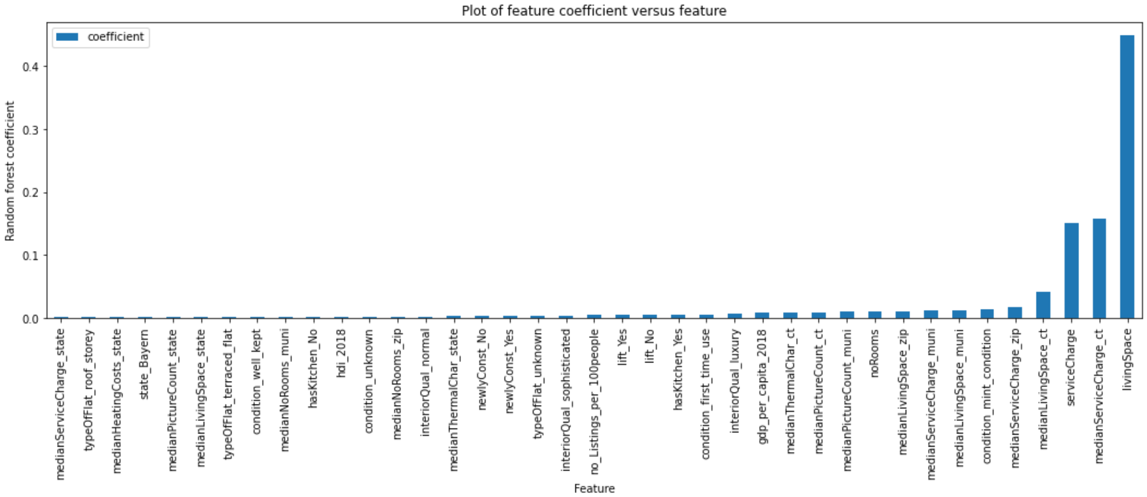


Figure 32: Features and their coefficient for the optimized random forest model

The top 10 features from the final random forest model were – living space, median city service charge, service charge, median city living space, median zip code service charge, mint condition, median municipality living space, median municipality service charge, median zip code living space, and the number of rooms. These features are similar to the ones observed from the linear models.

XGBoost algorithm

Initial XGBoost model and feature selection

The initial XGBoost model yielded a cross-validation r2 score of 0.84 ± 0.04 on the training set and an r2 score of 0.87 on the test set. A coefficient value of 0.001 was also chosen to pre-select the most important features. With feature selection, the test performance slightly dropped, with an r2 score of 0.86.

Hyperparameter tuning with feature selection

A new model was developed from the pre-selected features and the results of tuned parameters. The subsample proportion, column sample proportion and maximum depth of each tree were the parameters that were optimized using a random search algorithm on 4-fold cross validation set, with the mean squared error score as the assessment metric. The optimal parameters from the search grid were a subsample proportion of 0.7, a column sample proportion of 0.2 and a maximum tree depth of 4.

The eventual model yielded a slightly worse cross-validation r2 score of 0.83 ± 0.05 on the training set and an r2 score of 0.85 on the test set. Hyperparameter tuning and feature selection did not seem to improve the performance of the XGBoost algorithm.

The feature importance of the final model is shown in figure 33.

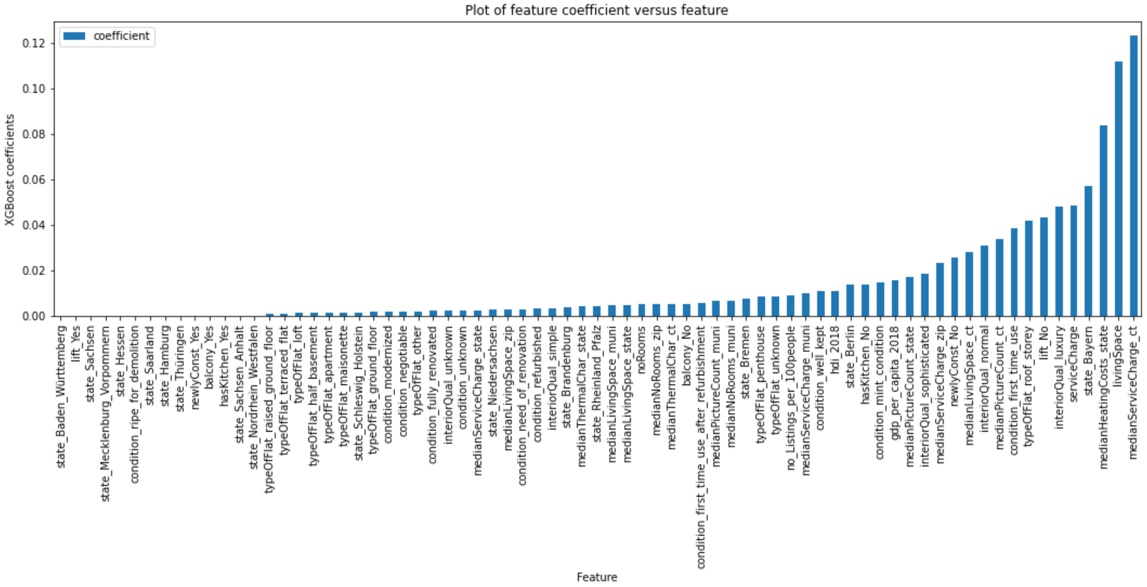


Figure 33: Features and their coefficient for the XGBoost model

The top 10 features from the original XGBoost model were – median city service charge, living space, median state heating cost, the state of Bayern, service charge, luxurious interior quality, no lift, roof storey flat, first-time-use condition, median city picture count.

The XGBoost model also produced similar top features as the other models, but also included features such as lift and roof storey, which were not observed in the previous models

Common features

All algorithms had the following common features in their top 20:

- Service charge
- Median city picture count
- Median zip code service charge
- Luxurious interior quality
- Living space
- Median city service charge
- Median city living space
- First-time-use condition

The only features of these common features that are controllable by a landlord are the service charge, the interior quality, the living space and the condition of first-time-use. In reality, the only variable that can be influenced is the interior quality.

MODEL SELECTION

The r2 score and mae score for the algorithms are shown in table 5 and table 6 below

Table 5: R-squared model performance summary

Model	Mean cv r2 score	Stdev of cv r2 scores	R2 test score
XGBoost	0.837	0.045	0.866
XGBoost w/ feat sel.	Null	Null	0.863
Random forest w/ hyper	0.819	0.047	0.861
XGBoost w/ hyper	0.827	0.047	0.849
Random forest w/ feat sel.	Null	Null	0.848
Random forest	0.807	0.049	0.841
Ridge	0.730	0.048	0.748
Lasso	0.730	0.048	0.747

Table 6: Mean absolute error model performance summary

Model	Mean cv mae score (€)	Stdev of cv mae scores (€)	mae test score (€)
Random forest w/ hyper	91.34	0.67	90.34
XGBoost	91.90	0.31	92.04
XGBoost w/ feat sel.	Null	Null	92.14
Random forest w/ feat sel.	Null	Null	97.19

Random forest	97.30	0.80	97.20
XGBoost w/ hyper	103.52	0.35	103.49
Lasso	145.63	0.81	146.49
Ridge	146.35	0.82	147.20

In general, the tree-induction algorithms performed better than the linear algorithms, suggesting that there were non-linear relationships in the dataset. The XGBoost model offered the best r-squared performance on both the training set with cross validation and the test set (0.866 and 0.837 respectively). It also had the lowest r-squared standard deviation of 0.045.

On the other hand, the random forest model with feature selection and hyper parameter tuning had the best mae score on both the test set and training set with cross validation (€90.34 and €91.34 respectively). This model performed worse than the XGBoost model in terms of the standard deviation of the mae score (€0.67 versus 0.31).

With a standard deviation of 0.67, the mae for the random forest model could get to as high as 92.01, while with a standard deviation of 0.31, the mae for the XGBoost model could get to as high as 92.21.

Both of these models are appropriate for any prediction tasks; however, the XGBoost model was selected as the preferred model because of the higher r2 scores and the lower standard deviation it offers for both the r2 and mae scores.

With the XGBoost model, the r2 score increased 0.75 to 0.866 (15.5% improvement) and the mae score increased from €147.29 to €92.04 (37.5% improvement).

Model application

The preferred model (XGBoost) was trained on the entire dataset, save the application set. The 5-fold cross-validation scores on this model was observed to be worse than cross validation scores observed over a train set, as shown in table 7 below.

Table 7: Performance summary of selected XGBoost model

Data	Mean cv r2 score	Stdev of cv r2 scores	Mean cv mae score (€)	Stdev of cv mae scores (€)
All	0.774	0.073	114.73	29.66
Training	0.837	0.045	91.90	0.31

These results were not entirely surprising, as different datasets were used. However, the surprising results were observed when the model was deployed on the application-set and compared the predicted rents to the original rents. Most of the original rents were set lower than the lowest rent predicted, as shown in table 8.

Table 8: Comparison of predicted and original rent

#	State	City/Town	Condition	rooms	Area (sqm)	Predicted rent (€)	Lower limit (€)	Set rent (€)	Higher limit (€)
1	Brandenburg	Uckermark	First time use	3.0	50.51	578.25	493.18	300.00	722.65
2	Schleswig Holstein	Dithmarschen	First time use after refurbishment	3.0	112.24	910.22	825.14	589.26	1054.61
3	Sachsen	Chemnitz	refurbished	2.0	51.67	641.11	556.03	518.00	785.50
4	Sachsen Anhalt	Halle Saale	negotiable	1.0	28.75	328.69	243.61	285.00	473.08
5	Hessen	Main Kinzig	Well kept	2.5	50.00	429.64	344.56	320.00	574.03

Of the five samples, only one property fell within the standard deviation of the mean absolute error scores for the model. This suggests that perhaps the four properties should increase their rents. Three of the properties have a condition that is either refurbished or of first-time use, suggesting that perhaps the landlords set low rents for these properties. However, the model's bias towards higher rents provides sufficient room for skepticism and further exploration of the data.

Conclusion and Recommendations

This project set out to analyze the factors that affect rent to inform the development of a model that predicts rent. There were clear associations between rent and certain features like the living area and service charge at different location levels. Certain categories, like the interior quality of the apartment also had a relationship with rent.

Machine learning models were developed to further unravel the relationships between rent and its features. The results from the models were consistent with the results from our exploratory analysis, wherein the living area, service charge and interior quality emerged as very important features.

The machine learning models were also essential in predicting rent. Two linear algorithms (ridge and lasso regression) were initially trialed to improve on the results of a baseline linear regression model, with an r-squared score of 0.75 and a mean absolute error score of

€147.29. They didn't. Tree-induction algorithms (random forest and XGBoost) were then trialed and succeeded in improving on the baseline performance - the best model being the XGBoost model with an r-squared score of 0.866 and a mean absolute error score of €90.34. It should be noted that the original dataset was divided into three: an application-set, a training set, and a test set. This division was necessary to assess model performance. Accordingly, the aforementioned models were trained on the training set and assessed on the test set.

Once, the final model was selected, the next step was to observe how the model performed in the wild: here is where the application set proved useful. The initial setback in terms of the final XGBoost model performance emerged when cross validation scores were obtained on the whole feature dataset (the training and test sets combined). With a mean cross validation performance of 0.774 and €114.73 for the r-squared and mean absolute error scores, expectations around model performance were tempered. Perhaps, this was expected as more data would necessarily yield different performance scores.

Nonetheless, using the model to predict rents for the application sets showed other kinks in model performance. Of the five samples, only the original rent of one of these samples fell within the margin of error predicted by the model. The model tended to overestimate the rent values. Perhaps, it was the case that landlords lowballed their rents, but the bias towards a higher estimate was concerning and required more investigation as part of another scope of work. That said, the performance on 5 samples of a dataset of over 263,000 samples is too small to reject the model.

To conclude, the project was successful in revealing patterns associated with rent and its features, and is likely useful in predicting rents with 'live' data.

Assumption, limitations and opportunities

- By selecting certain features over others during model development, it was assumed that the influence of these features in predicting rent was muted. Of relevance are all the date features that were excluded. Perhaps, this was a mistake and time-related features should have been included during model development.
- Inherent in using the rentals dataset are strong assumptions related to the quality of the data to justify any statistical inference. A distinct limitation of this project is the absence of any sampling methodology during information gathering. As mentioned, the dataset was obtained from Kaggle, and there was no opportunity to engage the data owner in terms of their sampling methodology. All that is known is that the data was retrieved from the Immoscout24 website on four days. Why these four days were chosen is unknown, especially in terms of periods between each retrieval date. Furthermore, a listing of over 263,000 rental records is impressive, but when compared to the population of Germany at over 83 million

people, there are questions regarding whether or not more information could be gathered from other sources - in order words, determining whether the sample of over 263,000 records is representative of the true character of all the rental apartments in Germany.

- Economic data was gathered from Wikipedia and on a state-wide basis. The state-wide information has likely changed, and regional information would be more useful.
- The text features of the dataset (description and facilities) offer opportunities for text analysis to unearth useful categories that influence rent.
- The hyperparameter tuning approach was limited due to the computational capabilities of the processor. Perhaps, better search methods (e.g., Bayesian optimization) and / or higher computational power offered opportunities to further refine models for increased performance.
- The whole dataset was analyzed, knowing fully well that regional disparities existed within the real estate industry in particular. A state-wide or city / town-wide approach to analyzing the data and developing models could have offered better modelling results. This shortcoming was slightly handled by augmenting the original dataset with aggregate features. But this approach has its limitations, because data was reduced to a statistic, while disregarding the associated relevant information from their distribution.
- A feature that would have been very useful is the amount of time a property has stayed on the market.
- This model will be challenged post-COVID. It is widely known that the real estate market has transformed in light of the pandemic. The last sample for the dataset was retrieved in February 2020, prior to COVID restrictions. A more recent dataset would be more appropriate for a post-COVID world.