

**A Scientific Paper**

**Rental Cost Prediction and Classification**

**Using K-Nearest Neighbours**

**Module: Data Science Foundations**

**Lena Moroz**

Middlesbrough, 2024

Table of Contents

[Abstract 3](#_Toc154877345)

[1. Introduction and Background 4](#_Toc154877346)

[1.1. Data Source 4](#_Toc154877347)

[1.2. Description 5](#_Toc154877348)

[2. Literature Review 7](#_Toc154877349)

[3. Technical Implementation (Developed Model) 8](#_Toc154877350)

[3.1. Data Science Processing Stages 8](#_Toc154877351)

[3.2. Data analysis using correlation and data visualisation 9](#_Toc154877352)

[3.3. Data preparation for training, validation and testing 12](#_Toc154877353)

[3.4. Machine learning model for classification, training, validation and testing procedures 13](#_Toc154877354)

[4. Performance Evaluation 14](#_Toc154877355)

[Findings and conclusions 18](#_Toc154877356)

[References 19](#_Toc154877357)

[APPENDIX 20](#_Toc154877358)

# **Abstract**

**Rental Cost Prediction and Classification Using K-Nearest Neighbours**

This study delves into the intricacies of a real estate dataset, employing correlation plots, scatter plots, and distribution analysis to unveil insights into the relationships between various property features and pricing. The findings indicate noteworthy connections between the number of bedrooms and bathrooms, as well as a robust correlation between accommodation size and bedroom/bathroom counts. Surprisingly, the accommodation's size exhibits a modest impact on its price, while a negative correlation is observed between price and the state it's located in.

Moreover, this business report explores the application of K-Nearest Neighbors (KNN) machine learning algorithm in predicting the category of rental costs in the USA real estate industry. The primary focus is on forecasting rental costs based on key factors such as square feet, location, and numerious amenities. The predictive model achieved an accuracy of 99%.

The report addresses several business questions to provide comprehensive insights into the rental market:

1. Variation in Classified Listings Categories:

Investigates how the category of classified listings varies based on attributes such as amenities, bathrooms, bedrooms, and square feet.

1. Relationship Between Rental Price and Features:

Examines the relationship between rental prices and various features like amenities, number of bathrooms, and bedrooms.

1. Influence of Location on Rental Prices:

Explores how the location influences rental prices and other characteristics of apartments in the dataset.

1. KNN for Identifying Similar Listings:

Investigates the feasibility of using KNN to identify similar apartment listings based on attributes like amenities, square feet, and bedrooms.

1. Insights from Attribute Relationships:

Explores the insights gained from analyzing the relationships between rental prices and other attributes in the dataset.

1. Variation in Apartment Characteristics Across Cities and States:

This report leverages the power of KNN to provide accurate predictions and valuable insights into the complex dynamics of the rental market. By addressing these business questions, stakeholders can make informed decisions, optimize property management strategies, and gain a deeper understanding of the factors influencing rental costs in the USA.

Our predictive model has achieved the high accuracy (99% for k=3). The model exhibits a commendable level of accuracy, indicating its ability to correctly classify price categories. This is particularly significant for business purposes where precise predictions are vital. While a higher k value (k=187) might suggest better generalisation, the lower accuracy and higher MSE indicate potential overfitting. The choice of k depends on the specific requirements of the business. However, regular monitoring and adaptation will contribute to the sustained success of the model in real-world applications.

# **Introduction and Background**

To begin with, the multifaceted landscape of rental real estate in the USA is a captivating subject of exploration in this research, driven by its intrinsic value as a rich dataset representing the intricacies of the real world. As a well-structured and meticulously collected compilation of data, the dataset serves as a treasure trove, enabling a granular analysis of the pivotal factors influencing rental costs in the real estate market.

This study unfolds against the backdrop of a burgeoning digital era where machine learning algorithms, particularly K-Nearest Neighbours (KNN), emerge as powerful tools for unravelling complex trends. By harnessing the potential of the KNN algorithm, we seek to discern the patterns governing rental costs in the USA real estate market. The dataset, having undergone meticulous preparation, is poised for analysis, with green-highlighted columns cleaned, processed, and converted into numeric formats for optimal compatibility with the model.

Moreover, this exploration is not only an academic endeavour but a practical resource for real estate agents and investors. The insights garnered from this survey hold the potential to inform critical decisions, offering a nuanced understanding of how various property features influence rental costs. As we embark on this journey, the fusion of real-world dataset exploration and advanced machine learning methodologies promises to contribute valuable perspectives to the realm of rental real estate decision-making.

## **Data Source**

Dataset Source(s): [**Apartments for rent**](https://archive.ics.uci.edu/dataset/555/apartment+for+rent+classified)

UC Irvine Machine Learning Repository. URL: <https://archive.ics.uci.edu/dataset/555/apartment+for+rent+classified>

Dataset: [**Apartments for rent**](https://archive.ics.uci.edu/dataset/555/apartment+for+rent+classified) **(2019)**

Industry: Real Estate.

Name of the Dataset: Apartment for rent classified.

This is a dataset of apartments for rent in the USA.

Instances: 10,000.

Features: 22.

Dataset contains diverse attributes such as location, size, amenities, associated costs and originates from the real-world dataset.

Reason for selection this dataset: with background in International Project Management and a strong interest in real estate, my purpose is to conduct an analysis of real estate market trends and patterns, and analyse the factors of forming the rental price.

**Real-World Relevance:** Understanding the cost of rent in different U.S. cities is a practical and real-world concern for many people, including students. It can be a relatable and relevant dataset for analysis and modelling.

**Data Exploration and Visualization:** Rental data provide opportunities to practice data exploration and visualisation techniques. One can create visualisations to compare rental prices in different locations.

**Predictive Modelling:** Rental data can also serve as a foundation for predictive modelling. One can build regression models to predict future rent prices based on various features like location, square footage and different amenities.

## **Description**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S/N** | **Column Name** | **Description** | **Data type before** | **Data type after** |
| 1. | id | unique identifier of apartment | numeric | deleted |
| 2. | category | category of classified | character | deleted |
| 3. | title | title text of apartment | character | deleted |
| 4. | body | body text of apartment | character | deleted |
| 5. | amenities | like AC, basketball, cable, gym, internet access, pool, refrigerator etc. | character | deleted |
| 6. | bathrooms | number of bathrooms | character | numeric |
| 7. | bedrooms | number of bedrooms | character | numeric |
| 8. | currency | price in current | character | deleted |
| 9. | fee | fee | character | deleted |
| 10. | has\_photo | photo of apartment | character | numeric |
| 11. | pets\_allowed | what pets are allowed dogs/cats etc. | character | numeric |
| 12. | price | rental price of apartment | integer | deleted |
| 13. | price\_display | price converted into display for reader | character | deleted |
| 14. | price\_type | price in USD | character | deleted |
| 15. | square\_feet | size of the apartment | integer | numeric |
| 16. | address | where the apartment is located | character | deleted |
| 17. | cityname | where the apartment is located | character | numeric |
| 18. | state | where the apartment is located | character | numeric |
| 19. | latitude | where the apartment is located | character | deleted |
| 20. | longitude | where the apartment is located | character | deleted |
| 21. | source | origin of data | character | deleted |
| 22. | time | when classified was created bout each attribute in the data set | integer | deleted |

Table 1: Variable before and after pre-processing steps

Data Preparation Key Points:

1. Green Highlighted Columns Processing for KNN:

* Columns highlighted in green underwent thorough cleaning and processing for compatibility with the K-Nearest Neighbors (KNN) algorithm.
* Conversion to numeric formats facilitated the integration of these features into the machine learning model.

1. Deletion of Original Category:

* The original category was removed from the dataset due to the presence of only one type of accommodation, rendering it non-contributory to the predictive model.

1. Handling of Title and Body Descriptions:

* Title and body columns, containing property descriptions, were retained and processed to extract relevant information that could contribute to the predictive model.

1. Complex Amenities Consolidation:

* The 'Amenities' column, featuring over 15 different characteristics for each property, was streamlined for analysis. The specifics of each amenity were not displayed individually but were considered collectively to enhance model efficiency.

1. Uniform Currency and Fee Values:

* Standardization was applied to ensure consistency, with all properties reflecting a currency of USD.
* A uniform fee value of 0 was assigned to each property, simplifying the dataset and eliminating potential noise.

1. Exclusion of Price Column at the Final Stage:

* The 'Price' column, initially part of the dataset, was strategically excluded in the final stages of data preparation. This step aligns with the transition to a categorical representation, optimizing the dataset for the KNN model.

1. Normalization of Time Data:

* Time-related data, initially in a non-readable condition, underwent normalization processes to enhance interpretability and facilitate meaningful analysis.

1. Introduction of 'Category' Column:

* To align with the predictive nature of the analysis, a new column, 'Category,' was introduced to replace the 'Price' column. This categorical representation serves as the target variable for the KNN model, reflecting the desired outcome of predicting rental cost categories.

These key points emphasize the meticulous data processing steps undertaken to ensure the dataset's readiness for KNN-based rental cost prediction. From cleaning and conversion to the strategic exclusion of redundant features, each step contributes to the optimization of the dataset for effective machine learning model implementation.

# **Literature Review**

The real estate industry, characterized by its non-linear economic system and complex structure, has seen a significant transformation with the advent of Big Data technologies. This literature review delves into key studies that explore the challenges and opportunities associated with collecting and analysing real estate data. There is some of the interesting abstracts about Big Data in Real Estate we found, which underline the specificity of this sphere and the problems of collecting and analysing real estate data.

**Determinants of Rental Rates in Major Cities in the United States (2022):** This study emphasizes the intricacies of determining rental rates in major U.S. cities. It identifies factors such as income levels, housing costs, and population densities as key determinants of rental rates. The research underscores the challenge of dealing with diverse city-level dynamics, emphasizing the need for localized analysis. Furthermore, it acknowledges the influence of outliers, such as rent control or free housing units, on national-scale data interpretation. This insight highlights the need for nuanced approaches in handling data variations across different urban landscapes. ([URL](https://digitalcommons.bryant.edu/eeb/vol3/iss1/6/))

**America’s Rental Housing (2022):** Focusing on the polarized nature of the rental market, this report highlights the stark divide between higher- and lower-income households. It notes the challenges faced by financially constrained households in finding affordable housing. Regional disparities in rent levels are attributed to differences in household incomes, land values, and the age of housing stock. The study sheds light on the urban-centric nature of the rental stock, emphasizing the importance of considering location-specific factors in the analysis of housing markets. ([URL](https://www.jchs.harvard.edu/sites/default/files/reports/files/Harvard_JCHS_Americas_Rental_Housing_2022.pdf))

**Real Estate Market Analysis System Based on Big Data (2023):** This research underscores the non-linear nature of the real estate industry and the need for continuous improvement in systems thinking. The study advocates for the construction of a dynamic database to capture the life cycle of houses. This approach enhances the application value of market analysis results in government decision-making. The paper emphasizes the normalization work required for monitoring and analyzing the real estate market. The incorporation of Big Data in real estate market analysis is seen as crucial for adapting to the industry's dynamic nature. ([URL](https://francis-press.com/uploads/papers/qbUS3ejFg0H36117tlc0Y3oLRTBe9WEhyfkf8pq8.pdf))

**Research on Real Estate Information System of the Real Estate Market Based on Big Data Technology (2021):** Focusing on the significance of a real estate information system, this article emphasizes the role of data efficiency and support for marketing strategies. The study delves into technical aspects, including system analysis, module design, structure planning, database design, and data collection methods. The creation of a comprehensive information system is highlighted as pivotal for supporting the smooth progress of the market. The article underscores the importance of collecting statistics on various real estate attributes, laying a foundation for effective system functioning. ([URL](https://www.e3s-conferences.org/articles/e3sconf/pdf/2021/33/e3sconf_aesee2021_02037.pdf))

In summary, these studies collectively highlight the specific challenges in the real estate industry and underscore the need for sophisticated approaches, particularly in handling diverse datasets and adapting to the non-linear dynamics of the market. The integration of Big Data technologies is identified as a crucial enabler for accurate market analysis and decision-making.

# **Technical Implementation (Developed Model)**

* 1. **Data Science Processing Stages**

1. Exploring dataset: this phase involves navigating through the well-structured and well-collected data, laying the foundation for subsequent analyses.
2. Selecting relevant columns for predicting the price: building upon the initial exploration, the dataset is refined by selecting specific columns deemed essential for predicting rental prices. This strategic curation ensures that the model focuses on the most influential features in the real estate market.
3. Outlier detection and deletion: rigorous checks are conducted to identify and eliminate outliers, ensuring that the dataset is free from anomalies that might skew the predictive model. This step contributes to the robustness and reliability of subsequent analyses.
4. Categorical to numeric conversion: categorical values within the dataset are transformed into numeric formats, a pivotal step for compatibility with the KNN algorithm. This conversion enhances the model's ability to discern patterns within the data.
5. Handling missing values: addressing missing values is imperative for a robust analysis. In this context, modes for bathrooms and bedrooms are calculated and used to replace missing values, fostering completeness in the dataset.

|  |  |
| --- | --- |
|  |  |
|  |  |

Table 2: Step by step data cleansing and preparation

* 1. **Data analysis using correlation and data visualisation**

A graph of a distribution of bathrooms

Description automatically generated

Fig. 1: Bar chart of distribution of bathrooms

A compelling bar chart illustrates the distribution of the 'bathrooms' variable in our dataset. Notably, there are over 30,000 properties with 1 bathroom, making it the most prevalent category. Following closely is the second most common configuration, featuring 3 bathrooms.

A graph of a number of bars

Description automatically generated

Fig. 2: Bar chart of distribution of bedrooms

This bar chart vividly illustartes that properties with 2 and 3 bedrooms are the most prevalent, with their combined count exceeding 40,000. In contrast, properties with other bedroom counts occur less frequently, highlighting the dominance of those with 2 and 3 bedrooms.

A graph of a number of apartments

Description automatically generated

Fig. 3: Apartment prices distribution

The analysis of Apartment Price Distribution reveals the discernible presence of outliers; however, the predominant spectrum of prices resides within the confines of 300 to 3000 dollars. Noteworthy is the apex frequency, where approximately 4000 instances of accommodation manifest at an equilibrium point of 1500 USD.

A graph with green and white squares

Description automatically generated

Fig. 4: Distribution of prices by the number of bedrooms

The distribution of prices by the number of bedrooms unveils a fascinating trend. Surprisingly, the highest prices are associated with properties featuring 7 bedrooms, surpassing even those with 8 or 9 bedrooms. This range fluctuates notably, spanning from 7 to 15 thousand USD. In contrast, the data underscores a substantial divergence for properties with 1 to 4 bedrooms.

A graph with orange dots

Description automatically generated

Fig. 5: Scatter plot of Square Feet vs. Price

The scrutiny of the scatter plot depicting square feet against price illuminates the presence of outliers and hints at a connection between property size and price. However, the correlation lacks substantial significance, given the prevalence of sizable accommodations with comparatively lower prices, as well as smaller lodgings with unexpectedly higher price tags.

A graph of a bar with a purple rectangle

Description automatically generated

Fig. 6: Bar chart of distribution of apartment categories

The examination of the distribution of apartment categories underscores a noteworthy observation: the dataset exclusively comprises a singular accommodation type, namely 'housing/rent/apartment.'

A screenshot of a computer screen

Description automatically generated

Fig. 7: Correlation plot

The correlation plot reveals notable connections between the number of bedrooms and bathrooms, indicating a reciprocal relationship. Similarly, a robust correlation is observed between the size of the accommodation and the counts of bedrooms and bathrooms. Moreover, the impact of the accommodation size on its price shows a relatively modest association, and there is a negative correlation between the price and the state it's located in. Interestingly, the presence or absence of photos for the accommodation seems to have no significant effect on any of the analysed parameters.

* 1. **Data preparation for training, validation and testing**

1. Creating rental cost categories:

- defining the breakpoints for categories,

- defining the labels for the categories,

- creating a new column with the categories.

A new column is created to represent these categories, setting the stage for predictive modelling.

1. Normalizing numeric features: numeric features are normalized to a common scale, mitigating disparities and ensuring that each feature contributes uniformly to the machine learning model.
2. Dataset splitting: the dataset is strategically divided into training and testing sets, a crucial step for evaluating the model's performance on unseen data.
3. Choosing machine learning approach: the features and the complexity of the data leads to the selection of KNN. This algorithm, known for its simplicity and effectiveness, aligns with the intricacies of the real estate dataset and the goals of predicting rental costs.
   1. **Machine learning model for classification, training, validation and testing procedures**

In the context of predicting rental prices, the idea that similar data points tend to have similar outcomes makes intuitive sense – properties in close proximity or with similar features are likely to have similar rental prices.

The main reasons for choosing k-Nearest Neighbours (KNN) as a machine learning model were:

KNN is a non-parametric algorithm, meaning it doesn't make assumptions about the underlying distribution of the data. This flexibility can be advantageous in scenarios where the relationship between features and rental prices is complex or not well-defined.

As KNN adapts well to local patterns in the data, it is valuable when predicting rental prices, where local market conditions and neighbourhood characteristics can strongly influence prices.

Unlike linear regression models, KNN doesn't assume a linear relationship between features and the target variable. This is beneficial when dealing with data where the relationship may not be strictly linear.

Machine Learning Model Configuration:

There was utilized the KNN algorithm for classification in the code. Before feeding the data into the model, I performed data normalization on all columns except the column representing the category of price, as it was kept in its original form.

The choice of normalization ensures that all features contribute equally to the distance calculations, an important aspect for KNN algorithms.

Data Splitting:

I divided the dataset into training and testing sets. The training set comprised 70% of the data, while the testing set contained the remaining 30%. This split was chosen to allow the model to learn patterns from the majority of the data and then evaluate its performance on unseen data.

The total number of observations in the dataset was 34,741. Remarkably, this number is approximately equal to the square of 186.39, leading to the selection of k = 186 and k =187 for the KNN model.

Odd Numbers for K:

It's important to note that KNN works optimally with odd values of k to avoid ties when voting. Therefore, even though we calculated accuracy for both =186 and k=187 will be used for the final model.

# **4. Performance Evaluation**

*Accuracy Calculation*

The accuracy of the k-Nearest Neighbours model was evaluated for two different values of k, specifically k = 186 and k =187. Accuracy represents the proportion of correctly classified instances in the test set.

For k = 186, the accuracy is 92.72666. It is calculated as the percentage of instances where the predicted labels match the true labels in the test set.

For k =187, a similar accuracy calculation is performed. The accuracy is equal to 92.69308.

The accuracy for k=3 is 99.43586.

*Tabular Representation of Predictions*

The model's predictions were compared against the actual values in tabular form for both k = 186 and k =187. These tables provide a clear breakdown of correct predictions and misclassifications, allowing for a detailed analysis of the model's performance.

A screen shot of a computer

Description automatically generated

Fig. 8: Prediction against actual value for k=186

A screenshot of a computer screen

Description automatically generated

Fig. 9: Prediction against actual value for k=187

*Confusion Matrix*

Two confusion matrices were constructed using the caret package for k=187 and k=3. This type of matrix offers a comprehensive view of the model's classification performance, detailing true positives, true negatives, false positives, and false negatives.

A graph with different colored squares

Description automatically generated

Fig. 10: Confusion Matrix for k=187

A graph with different colored squares

Description automatically generated

Fig. 11: Confusion Matrix for k=3

*Optimization*

An optimization loop was implemented to determine the optimal value of k for the KNN model. The loop iterated through values of k from 1 to 187, calculating the accuracy for each k and storing the results in the k.optm vector.

The final output identifies the value of k that yields the highest accuracy on the test set.

The highest accuracy was achieved for k=1. It was equal to 99.56347.

A graph of a graph

Description automatically generated with medium confidence

Fig. 12: Plot of accuracy for k=1 to k=187

*Statistical Tests*

We will use the Mean Squared Error to give us an idea of how well our model is performing in terms of the squared differences between predicted and actual values. Lower MSE values indicate better model performance, as they signify smaller deviations between predicted and actual values.

Mean Squared Error: 0.07306917 (for k=187).

Mean Squared Error: 0.00564137 (for k=3).

Comparing these values, the model with k=3 exhibits a significantly lower MSE, suggesting that it performs better in minimizing the squared differences between predicted and actual values. This indicates that the model with k=3 provides more accurate predictions and is a preferred choice over the one with k=187 in terms of minimizing prediction errors.

Area Under the Curve: 0.6175 for k=187. An AUC value of 0.6174545 indicates a moderate level of discriminatory power. The ROC curve suggests that the model performs better than random chance but may benefit from further improvement.

AUC: 0.5968725 for k=3. While the model demonstrates a fair level of discriminative ability with an AUC value of 0.5968725, there is room for improvement.

A graph of a curve

Description automatically generated

Fig. 13: Receiver Operating Characteristic for k=187

An Area Under the Curve value of 0.5968725 for k=3 suggests a moderate level of performance for the model in predicting the category of price. An AUC value closer to 1.0 indicates better discrimination, while a value around 0.5 suggests random chance. In the case of 0.5968725, the model demonstrates some ability to distinguish between categories, but there is room for improvement. Further analysis and potential model enhancements may be explored to boost the AUC value, leading to more reliable predictions and better discrimination between different price categories.

# **Findings and conclusions**

The significance of the results obtained in the evaluation of the k-nearest neighbors (k-NN) model is crucial for understanding the performance and reliability of the predictive system. The discussion should encompass various aspects, including accuracy, Mean Squared Error (MSE), and considerations related to commercial risks, legal aspects, and privacy issues.

The prediction results for the category of price with k = 187 achieved an accuracy of 92.69308. This indicates a high level of accuracy in the model's ability to correctly classify the price categories. A higher accuracy percentage suggests that the model's predictions closely align with the actual categories, providing confidence in its effectiveness. The outcome of 92.69308 indicates that nearly 93% of the predictions were correct, showcasing the robust performance of the model in categorizing prices with the chosen value of k.

However, we have got higher accuracy for k=3, wich is equal to 99%. After taking some statistical tests we can conclude that some tests show the better performance of the model for k=3 (for example, in Mean Squared Error), and another shows that model with k=187 performs better in the Area Under the Curve test.

Choosing the optimal value for k in k-nearest neighbors (k-NN) involves a trade-off between accuracy and model complexity. Analysing the results of accuracy calculation and statictical tests, it appears that k=3 outperforms k=187 in terms of accuracy and MSE. However, it's important to note that the choice of k can depend on the nature of our data.

If a higher accuracy (99%) is crucial for the application, k=3 seems to be the better choice. Sometimes, a more complex model (higher k) might generalize better, but in this case, the simpler model (lower k) performs better according to accuracy and MSE.

Moreover, we should take into account, that provided results were made on the base of test data and running the model on new real-world data might be with different outcomes. We should consider that model with k=3 will be more sensitive for outliers, which are common in the real estate sphere. Thus, k=187 appears to be the better choice for data with a large spread. Another decision could be taking the k number between 3 and 187 to mix the pros and cons of the two models. All in all, it's recommended to perform further analysis and possibly conduct cross-validation to ensure the robustness of the model's performance.

Before deploying the model in a commercial setting, it's crucial to assess its robustness and reliability. The higher accuracy and lower MSE for k=3 suggest that this configuration might be more suitable for deployment. Considerations should be given to the scalability of the model. A simpler model (lower k) might be computationally less intensive, making it more scalable for larger datasets or real-time applications.

The data sourced from the UCI Repository doesn’t introduce any considerations related to data privacy. But for further work with real estate data, we should compliance with data protection regulations governing the use of the dataset.

In conclusion, the results suggest that k=3 is a favorable choice based on higher accuracy and lower MSE. However, a comprehensive evaluation should include considerations related to commercial risks, legal compliance, and privacy issues to ensure responsible and ethical deployment of the predictive model. Regular monitoring and adaptation to changing conditions will be essential for the model's sustained success.

# **References**

1. Apartment for Rent Classified (no date) UCI Machine Learning Repository. Available at: https://archive.ics.uci.edu/dataset/555/apartment+for+rent+classified (Accessed: 21 November 2023).
2. Available Cran packages by name (no date) CRAN Packages By Name. Available at: https://cran.r-project.org/web/packages/available\_packages\_by\_name.html#available-packages-E (Accessed: 21 November 2023).
3. Grolemund, H.W. and G. (no date) R for data science, Welcome. Available at: https://r4ds.had.co.nz/index.html (Accessed: 21 November 2023).
4. Kobzan, S. and Pomortseva, O. (1970) Factors of influence on the real estate market, SpringerLink. Available at: https://link.springer.com/chapter/10.1007/978-3-031-31248-9\_5 (Accessed: 21 November 2023).
5. RStudio open-source packages (no date) RStudio. Available at: https://www.rstudio.com/about/rstudio-open-source-packages/ (Accessed: 21 November 2023).
6. ‘Real estate market analysis system based on Big Data’ (2023) Academic Journal of Computing; *Information Science*, 6(7). doi:10.25236/ajcis.2023.060723. (Accessed: 21 November 2023).
7. Li, B. (2021) ‘Research on real estate information system of the real estate market based on Big Data Technology’, *E3S Web of Conferences*, 257, p. 02037. doi:10.1051/e3sconf/202125702037. (Accessed: 21 November 2023).
8. Ruivo, R. (2010). Determinants of Rental Rates in Major Cities in the United States. Empirical Economic Bulletin, An Undergraduate Journal, [online] 3(1). Available at: <https://digitalcommons.bryant.edu/eeb/vol3/iss1/6/>. (Accessed: 21 November 2023).
9. America’s Rental Housing 2022 | Joint Center for Housing Studies. [online] Available at: <https://www.jchs.harvard.edu/americas-rental-housing-2022>. (Accessed: 21 November 2023).
10. ‘Real estate market analysis system based on Big Data’ (2023) Academic Journal of Computing; *Information Science*, 6(7). doi:10.25236/ajcis.2023.060723.
11. Greene, D. (2022) ‘Landlords of the internet: Big data and big real estate’, *Social Studies of Science*, 52(6), pp. 904–927. doi:10.1177/03063127221124943.
12. Mourouzi-Sivitanidou, R. and Sivitanides, P. (2020) ‘Data sources for real estate market analysis’, *Market Analysis for Real Estate*, pp. 427–432. doi:10.1201/9780429279409-26.

APPENDIX

Source Code

install.packages(c("caret", 'modeest', 'crayon', 'tidyverse', 'dplyr', 'ggplot2', 'lattice', 'kernlab', 'e1071'))

install.packages('ggplot2')

install.packages("class")

library(caret)

library(dplyr)

library(modeest)

library(crayon)

library(tidyverse)

library(ggplot2)

library(lattice)

library(e1071)

library(kernlab)

getwd()

apartments <- read.csv('apartments\_for\_rent\_classified\_100K.csv', sep=';', header=TRUE)

summary(apartments)

str(apartments)

head(apartments)

#Exploring dataset with ggplot

#Histogram of Prices

ggplot(apartments, aes(x = as.numeric(price))) +

geom\_histogram(binwidth = 100, fill = "blue", color = "black") +

labs(title = "Distribution of Apartment Prices",

x = "Price (USD)",

y = "Frequency")

#Boxplot of Prices by Bedrooms

ggplot(apartments, aes(x = bedrooms, y = as.numeric(price))) +

geom\_boxplot(fill = "lightgreen", color = "darkgreen") +

labs(title = "Prices Distribution by Number of Bedrooms",

x = "Number of Bedrooms",

y = "Price (USD)")

#Scatter Plot of Square Feet vs. Price

ggplot(apartments, aes(x = as.numeric(square\_feet), y = as.numeric(price))) +

geom\_point(color = "orange") +

labs(title = "Scatter Plot of Square Feet vs. Price",

x = "Square Feet",

y = "Price (USD)")

#Bar Chart of Apartment Categories

ggplot(apartments, aes(x = category)) +

geom\_bar(fill = "purple", color = "black") +

theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +

labs(title = "Distribution of Apartment Categories",

x = "Apartment Category",

y = "Count")

#Remove Invalid Values

apartments <- na.omit(apartments)

#Processing and Pre-processing

library(dplyr)

#select appropriate columns for analysis

rent <- apartments %>% select(6:7, 10:12, 15, 17, 18)

summary(rent)

str(rent)

#see all unique values from one column

unique\_values <- unique(rent$pets\_allowed)

unique\_values

#count the occurrences of each unique value in a column to replace missing values with the most common

count\_values <- table(rent$pets\_allowed)

count\_values

# convert to categories

rent$pets\_allowed[rent$pets\_allowed == "None"] <- 0

rent$pets\_allowed[rent$pets\_allowed == "Cats,Dogs"] <- 1

rent$pets\_allowed[rent$pets\_allowed == "Dogs"] <- 1

rent$pets\_allowed[rent$pets\_allowed == "Cats"] <- 1

rent$pets\_allowed[rent$pets\_allowed == "Cats,Dogs,None"] <- 1

rent$pets\_allowed[rent$pets\_allowed == "null"] <- 0

#Cleaning data

rent <- rent %>% filter(!pets\_allowed %in% c(1242, 1252, 1299, 1325, 1545, 1601, 1622, 1900, 1930, 2305, 2375, 3010, 575))

# check the data type

str(rent)

unique\_values <- unique(rent$pets\_allowed)

unique\_values

rent$pets\_allowed[rent$pets\_allowed == "No"] <- 0

rent$pets\_allowed[rent$pets\_allowed == "Thumbnail"] <- 1

rent$pets\_allowed[rent$pets\_allowed == "Yes"] <- 1

rent$pets\_allowed <- as.numeric(rent$pets\_allowed)

#change data type

rent$pets\_allowed <- as.numeric(rent$pets\_allowed)

rent$bathrooms <- as.numeric(rent$bathrooms)

rent$bedrooms <- as.numeric(rent$bedrooms)

rent$price <- as.numeric(rent$price)

rent$square\_feet <- as.numeric(rent$square\_feet)

view(rent)

anyNA(rent)

na\_counts <- colSums(is.na(rent))

#show the number of NAs for each column

na\_counts

rent <- na.omit(rent)

anyNA(rent)

# Calculate mode

mode\_pets <- as.integer(modeest::mfv(rent$pets\_allowed))

mode\_pets

rent$pets\_allowed <- ifelse(rent$pets\_allowed == 'NA', mode\_pets, rent$pets\_allowed)

table(rent$pets\_allowed)

#view(rent)

str(rent)

unique\_values2 <- unique(rent$bathrooms)

unique\_values2

#delete rows with float numbers in the column of bathrooms

rent <- rent %>% filter(!bathrooms %in% c(1.5, 2.5, 3.5, 4.5, 5.5, 8.5))

mode\_bath <- as.integer(modeest::mfv(rent$bathrooms))

# replacing missing values with mode

rent$bathrooms[is.na(rent$bathrooms)] <- mode\_bath

table(rent$bathrooms)

table(rent$bedrooms)

table(rent$has\_photo)

rent$has\_photo[rent$has\_photo == "No"] <- 0

rent$has\_photo[rent$has\_photo == "Thumbnail"] <- 1

rent$has\_photo[rent$has\_photo == "Yes"] <- 1

table(rent$has\_photo)

rent$has\_photo <- as.numeric(rent$has\_photo)

view(rent)

str(rent)

# create categories of rental cost

# Define the breakpoints for categories

breaks <- c(-Inf, 1000, 2000, Inf)

# Define the labels for the categories

labels <- c("Low", "Medium", "High")

# Create a new column with the categories

rent$category <- cut(rent$price, breaks = breaks, labels = labels, include.lowest = TRUE)

# Convert specific columns to categorical

#rent <- as.data.frame(lapply(rent, as.factor))

columns\_to\_convert <- c("cityname", "state", "category")

#define a function to convert columns to factors without changing values

convert\_to\_factor <- function(x) {

if(is.numeric(x) | is.character(x)) {

as.factor(x)

} else {

x

}

}

# Apply the custom function to specified columns

rent[, columns\_to\_convert] <- lapply(rent[, columns\_to\_convert], convert\_to\_factor)

str(rent)

view(rent)

rent$has\_photo <- as.numeric(as.character(rent$has\_photo))

rent$pets\_allowed <- as.numeric(as.character(rent$pets\_allowed))

rent\_numeric <- as.data.frame(lapply(rent, as.numeric))

#view(rent\_numeric)

#remove the column of actual price

rent\_num <- rent\_numeric %>% select(1:4, 6:9)

view(rent\_num)

#check the column of square feet for outliers

boxplot(rent$square\_feet, main="Boxplot of Square Feet")

# Calculate summary statistics

summary\_stats <- summary(rent\_num$square\_feet)

print(summary\_stats)

#Delete rows where 'square\_feet' is less than 100

rent\_num <- rent\_num[rent\_num$square\_feet >= 100, ]

# Identify potential outliers using the IQR method

#rent\_num$square\_feet <- as.numeric(as.character(rent\_num$square\_feet))

#Q1 <- quantile(rent\_num$square\_feet, 0.25)

#Q3 <- quantile(rent\_num$square\_feet, 0.75)

#IQR <- Q3 - Q1

#lower\_bound <- Q1 - 1.5 \* IQR

#upper\_bound <- Q3 + 1.5 \* IQR

#potential\_outliers <- rent\_num$square\_feet[rent\_num$square\_feet < lower\_bound | rent\_num$square\_feet > upper\_bound]

# Display potential outliers

#if (length(potential\_outliers) > 0) {

# cat("Potential outliers:\n")

# print(potential\_outliers)

#} else {

# cat("No potential outliers found.\n")

#}

#Checking and comleting missing values

missing\_rent <- sum(!complete.cases(rent\_num))

cat("Number of missing values in dataset:", missing\_rent, "\n")

# Remove rows with missing values

#rent\_num <- rent\_num[complete.cases(rent\_num), ]

correlation\_matrix <- cor(rent\_num)

ggplot(data = as.data.frame(as.table(correlation\_matrix)),

aes(x = Var1, y = Var2, fill = Freq)) +

geom\_tile(color = "white") +

scale\_fill\_gradient2(low = "blue", mid = "white", high = "red",

midpoint = 0, limits = c(-1, 1)) +

theme\_minimal() +

labs(title = "Correlation Plot")

#barplot(table(rent\_num$bathrooms), main = "Distribution of Bathrooms")

#barplot(table(rent\_num$bedrooms), main = "Distribution of Bedrooms")

# Data Normalization

head(rent\_num)

normalize <- function(x) {

return((x-min(x))/(max(x) - min(x))) }

#normalize each column separately

#do not normalize the target feature of price category

rent\_norma <- rent\_num[, c(1:7)]

rent\_norma <- as.data.frame(lapply(rent\_norma, scale))

head(rent\_norma)

rent\_norma$category <- rent\_num$category

rent\_norma

# Data Splicing

set.seed(123)

data <- sample(1:nrow(rent\_norma), size=nrow(rent\_norma)\*0.7, replace = FALSE)

train.rent <- rent\_norma[data,]

test.rent <- rent\_norma[-data,]

#Creating separate dataframe for 'category' feature which is our target

train.rent\_labels <- rent\_norma[data,8]

test.rent\_labels <- rent\_norma[-data,8]

library(class)

#Find the number of observation

NROW(train.rent\_labels)

result <- sqrt(34741)

result

knn.186 <- knn(train=train.rent, test=test.rent, cl=train.rent\_labels, k=186)

knn.187 <- knn(train=train.rent, test=test.rent, cl=train.rent\_labels, k=187)

knn.3 <- knn(train=train.rent, test=test.rent, cl=train.rent\_labels, k=3)

#Model Evaluation

#Calculate the proportion of correct classification for k = 186, 187

ACC.186 <- 100 \* sum(test.rent\_labels == knn.186)/NROW(test.rent\_labels)

ACC.187 <- 100 \* sum(test.rent\_labels == knn.187)/NROW(test.rent\_labels)

ACC.3 <- 100 \* sum(test.rent\_labels == knn.3)/NROW(test.rent\_labels)

ACC.186

ACC.187

ACC.3

# Check prediction against actual value in tabular form for k=186

table(knn.186 ,test.rent\_labels)

knn.186

# Check prediction against actual value in tabular form for k=187

table(knn.187,test.rent\_labels)

table(knn.3 ,test.rent\_labels)

library(caret)

library(ggplot2)

confusionMatrix(table(knn.187 ,test.rent\_labels))

cm <- confusionMatrix(table(knn.187, test.rent\_labels))

conf\_matrix <- as.matrix(cm)

# Convert to data frame for ggplot

conf\_df <- as.data.frame(conf\_matrix)

conf\_df$Actual <- rownames(conf\_df)

# Reshape the data for ggplot

conf\_df\_long <- tidyr::gather(conf\_df, Predicted, Count, -Actual)

# Create a ggplot bar chart

ggplot(conf\_df\_long, aes(x = Actual, y = Count, fill = Predicted)) +

geom\_bar(stat = "identity") +

labs(title = "Confusion Matrix", x = "Actual", y = "Count") +

theme\_minimal()+

theme(plot.title = element\_text(hjust = 0.5))

#Optimization

i=1

k.optm=1

for (i in 1:187){ knn.mod <- knn(train=train.rent, test=test.rent, cl=train.rent\_labels, k=i)

k.optm[i] <- 100 \* sum(test.rent\_labels == knn.mod)/NROW(test.rent\_labels)

k=i

cat(k,'=',k.optm[i],'')}

#The accuracy for k=1 is the best: 1 = 99.56347

#The accuracy for k=187 is 92.69308

#The accuracy for k=3 is 99.43586.

#represent accuracy graphically

plot(k.optm, type="b", xlab="K- Value",ylab="Accuracy level")

#Statictical tests

# Convert factors to numeric

actual\_values <- as.numeric(as.character(test.rent\_labels))

predicted\_values <- as.numeric(as.character(knn.187))

# calculate the Mean Squared Error for test results

mse <- mean((actual\_values - predicted\_values)^2)

# Print the MSE

cat("Mean Squared Error:", mse)

install.packages("pROC")

install.packages("ROCR")

library(pROC)

library(ROCR)

# Create an ROC curve

actual\_values <- as.numeric(as.character(test.rent\_labels))

predicted\_values <- as.numeric(as.character(knn.3))

roc\_curve <- roc(ifelse(actual\_values == 2, 1, 0), predicted\_values)

plot(roc\_curve, main = "ROC Curve", col = "blue", lwd = 2)

# Calculate AUC (Area Under the Curve)

auc\_value <- auc(roc\_curve)

cat("AUC:", auc\_value, "\n")

# Print the ROC curve details

print(roc\_curve)

# Combine actual and predicted values into a dataframe

distribution\_data <- data.frame(Category = c(rep("Actual", length(actual\_values)), rep("Predicted", length(predicted\_values))),

Value = c(actual\_values, predicted\_values))

# Create a bar plot for class distribution

class\_distribution\_plot <- ggplot(distribution\_data, aes(x = Category, fill = as.factor(Value))) +

geom\_bar(position = "dodge") +

labs(title = "Actual vs. Predicted Class Distribution", x = "Category", y = "Count") +

scale\_fill\_manual(values = c("Actual" = "blue", "Predicted" = "red")) + # Adjust colors as needed

theme\_minimal()

# Display the plot

print(class\_distribution\_plot)