Technical University of Denmark

Data: Feature extraction, and visualization
Project 1

October 3, 2023

Author:

Zhenlin Xie	s232268
Andro Kranjcevic	s204704
João Luís Gonçalves Mena	s223186

Technical University of Denmark
Course: 02450 - Introduction to Machine Learning and Data Mining

Contents

Contributions Table			
	Zhenlin	Andro	Joao
Section 1	0%	100%	0%
Section 2	7.5%	80%	12.5%
Section 3	0%	0%	100%
Section 4	50%	25%	25%
Exam Problems	33.33%	33.33%	33.33%

1 Introduction

Airbnb operates an online marketplace for short term rentals, and acts as a broker by charging a commission from each booking. Analyzing data from Airbnb rentals can assist prospective renters in making decisions about rentals, for example, determining an appropriate pricing level, minimum nights, or location. For this project, the data is collected through Inside Airbnb (http://insideairbnb.com/get-the-data/). The data corresponds to the city of Copenhagen, containing information from 17028 rentals currently listed in Copenhagen. In total there are 17 features that provide great insight of the data, such as neighbourhood, room type, price, period of stay, etc.

In a study by Rezazadeh et al. [2], a price prediction model used machine learning, deep learning, and NLP techniques. Support Vector Regression (SVR) outperformed linear regression, tree-based models, and neural networks, showing the lowest Mean Squared Error and highest R^2. Dhillon et al. [1] employed Linear Regression, Logistic Regression, and Random Forest for Airbnb listing price prediction. Random Forest had the lowest Root Mean Squared Error. Both papers followed common data transformation steps: standardization, outlier removal, and handling missing values.

In regression analysis, the aim is to create a model that can predict prices for a wide range of listings based on the patterns it has learned from the data. By utilizing the most significant features as input variables, the exact or continuous price (output variable) for each listing will be estimated through regression. When using our classification algorithm on Airbnb data, the aim is to predict classes based on data probabilities. In Binary classification, 'low' or 'high' rental prices will be predicted, indicating if they're below or above the median. In Multiclass, the 'medium' will be added for prices between the 33rd and 66th percentile. Data transformation is crucial in both classification and regression, involving standardization, feature selection, and handling missing values.

2 Data Attributes

The initial dataset version contained 17 attributes. However, due to the lower importance of certain attributes and their interrelation, they have been removed.
There is missing data in the date column due to the fact that there are no reviews for that certain period. Since license and neighbourhood group don't have any values, they will simply be discarded. Columns id, host_id, last_review, reviews_per_month, number_of_reviews_ltm, calculated_host_listings_count, and host_name will also be discarded since that information isn't useful for our analysis and will have no impact on the result.
Additionally, the data parsing from the name column into separate columns named 'rating', 'bedroom' and 'bath' will be performed. Having completed all the necessary steps, we are now left with 11 attributes.

neighbourhood	latitude	longitude	room_type	price	min_nights	reviews	avail_365	rating	bedroom	bath
...	...	...	...	...	...	...	...	...	...	...

Table 1: Data Attributes

To describe the attributes more precisely, they will be categorized as discrete or continuous and as nominal, ordinal, interval, or ratio:

neighbourhood - Discrete and Nominal, **latitude** - Continuous and Interval, **longitude** - Continuous and Interval

room_type - Discrete and Nominal, **price** - Continuous and Ratio, **min_nights** - Discrete and Ratio

reviews - Continuous and Ratio, **availability_365** - Discrete and Ratio, **rating** - Continuous and Ordinal

bedroom - Discrete and Ratio, **bath** - Discrete and Ratio

The next logical step is to delve into the dataset's statistics, as this will aid in gaining a deeper understanding of its characteristics. Furthermore, it will assist us in identifying any missing values, outliers, or anomalies, and may even spark ideas for feature engineering. It's important to note that, owing to their respective string and datetime data types, they will be omitted from the summary statistics. Instead, the focus will exclusively remain on describing the int and float data types.

Table 2: Descriptive statistics for the dataset

	price	minimum_nights	number_of_reviews	availability_365	rating	bedroom	bath
Count	17026	17026	17026	17026	12004	17023	15405
Mean	1261.13	4.51	17.45	85.99	4.70	1.64	1.37
Std	1868.71	15.43	39.02	115.19	0.65	0.91	1.08
Min	120	1	0	0	0	1	0
25%	799	2	1	0	4.70	1	1
50%	1000	3	6	18	4.86	1	1
75%	1401	4	18	160	5	2	1
Max	150364	1111	1178	365	5	15	5

Observations reveal that, on average, the minimum number of nights required for booking is approximately 4.5, indicating that most listings have a relatively short minimum stay requirement. These listings are available for about 86 days out of the year, implying significant occupancy throughout the year. Additionally, the average number of reviews suggests that properties receive a moderate level of feedback from guests. The most notable anomaly is the maximum price of 150,364 DKK per night, which suggests the possibility of incorrect data entry. Furthermore, listings that show no availability throughout the year may warrant investigation. This absence of availability could be attributed to permanent occupation or non-standard agreements. Some listings haven't received any reviews, but since it's not a significant unknown, those observations will not be removed from the dataset.

It is important to investigate to which extent the features correspond to the target variable (price). Because of that, most of the interest lies within the first row of a map, since the price is the variable that should be predicted. The correlation coefficient is calculated to estimate the relationship between the selected features. The Pearson correlation coefficient is used in this analysis. If both variables rise or fall simultaneously, then the correlation coefficient is positive, and approaching the value of 1, and vice versa (towards -1). The correlation coefficient is

calculated between the features, and the strength of linear dependence between them is visualized with a heat map. The light color indicates a high correlation, while the dark color indicates the opposite, as visible in the figure 1 below.

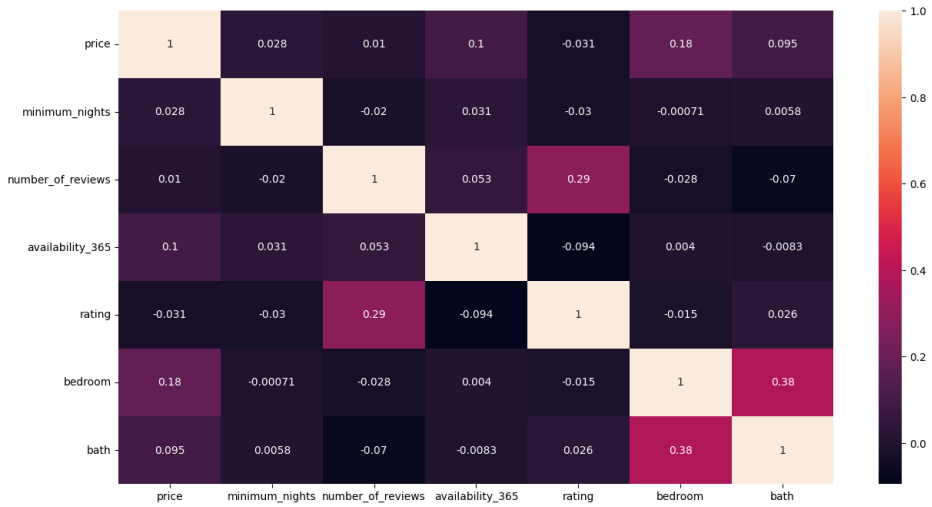


Figure 1: Pearson Correlation Map

In principle, the attributes appear to have a low correlation with the target variable, price, with most coefficient values around 0. As expected, the highest correlation is between 'bath' and 'bedroom' attributes, while the lowest (negative) correlation is seen between 'rating' and 'bath' attributes, indicating that as the number of bathrooms increases, the rating tends to decrease.

3 Data Visualization

In this section, a series of data visualizations regarding the dataset is, in order to help comprehend the characteristics of the data and understand if the attributes appear to be normal distributed, if there are any issues with outliers in the data, if the variables are correlated and how feasible does our primary machine learning model objective appear to be. When generating these visualizations, the ACCENT principles and Tufte's guidelines were taken into account. This will allow for more clear and insightful visualizations, enabling a better communication of the reasoning process to the reader.

Since the aim of the model is to perform predictions regarding prices, the attribute 'price' will be the spotlight in our dataset. By looking at the table **??**, it is noticeable that the standard deviation on the price is very high, and the maximum value is orders of magnitude above most of the values, which is not realistic. Looking at the following boxplot, the prices appear very unbalanced due to the outliers:

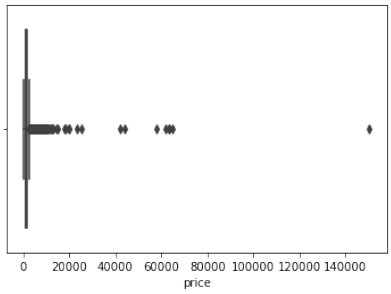


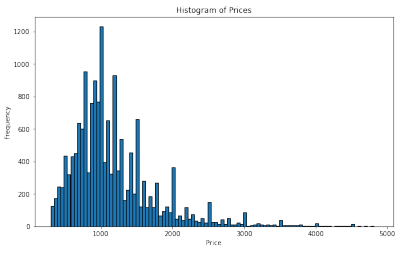
Figure 2: Horizontal Boxplot of the prices

To overcome the problems that these outliers may bring, the dataset will be filtered by price to the entries with values between the 1st and 99th percentiles, removing the extreme cases. The 'minimum_nights' and 'number_of_reviews' attributes also presented some incomprehensible standard deviations and max values. After some reflection, it was decided that it is normal that the 'number_of_reviews' may differ largely for some of the most and less popular places, but having '1111' minimum nights is not realistic and may have been caused by entering a random value or there may have been an input error, for example. For this reason, the same filtering was applied to the attribute 'minimum_nights' as it was with 'price'. With these changes, the descriptive statistics now have the following appearance:

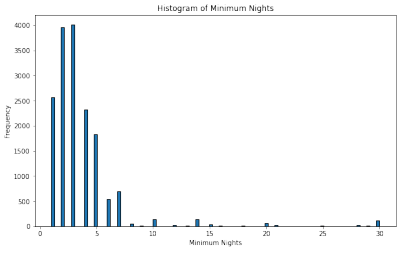
Table 3: Descriptive statistics for the dataset after filtering prices

	latitude	longitude	price	minimum_nights	number_of_reviews	availability_365	rating	bedroom	bath
count	16584.00	16584.00	16584.00	16584.00	16584.00	16584.00	11763.00	16368.00	15122.00
mean	55.68	12.56	1178.02	3.64	17.42	85.24	4.70	1.64	1.37
std	0.02	0.03	608.77	3.48	38.57	114.67	0.65	0.91	1.08
min	55.62	12.46	300.00	1.00	0.00	0.00	0.00	1.00	0.00
25%	55.67	12.54	800.00	2.00	2.00	0.00	4.70	1.00	1.00
50%	55.68	12.56	1000.00	3.00	6.00	18.00	4.86	1.00	1.00
75%	55.70	12.58	1400.00	4.00	18.00	157.00	5.00	2.00	1.00
max	55.73	12.64	4857.00	30.00	1178.00	365.00	5.00	15.00	5.00

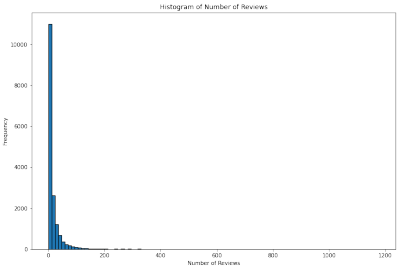
After applying the filters, the values look more stable. An histogram of these main attributes was plotted to verify if they are normal distributed.



(a) Prices histogram



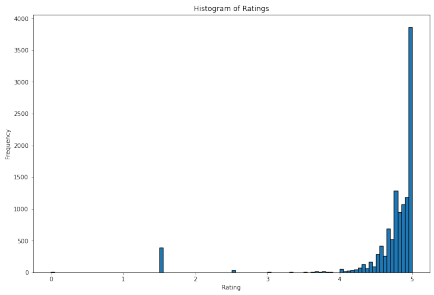
(b) Minimum nights histogram



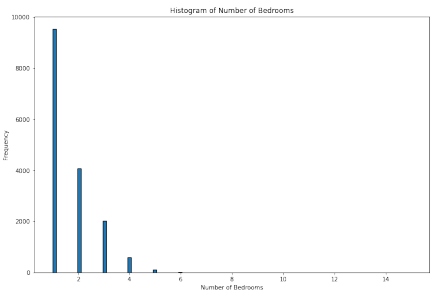
(c) Number of reviews histogram

According to the generated plots, the 'prices' and the 'minimum_nights' appear to be approximately normally distributed, which is a good indicator. As expected, the 'number_of_reviews' presents an asymmetric distribution.

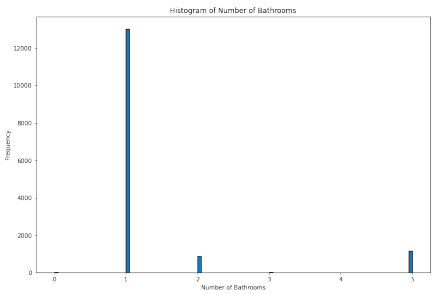
Shifting to other attributes, most of the 'rating' values are between 4 and 5, and even though there is some asymmetry in the distribution, this shouldn't be a concern, and no values should be considered outliers here. The 'bedrooms' and 'bath' histograms also show skewed distributions, but the values appear reasonable and realistic, and it makes sense that most places will have a low number of bedrooms and bathrooms, but it is viable to have as much as 15 bedrooms and 5 bathrooms, which are the highest values in our dataset.



(a) Ratings histogram

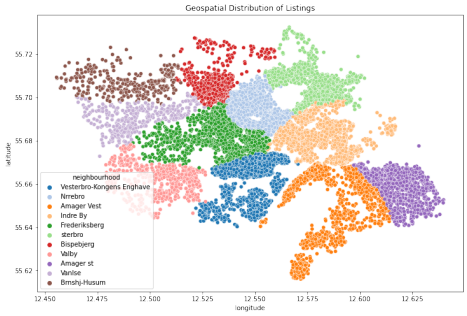


(b) Bedrooms histogram

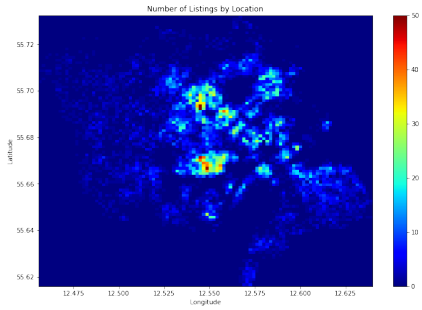


(c) Bathrooms histogram

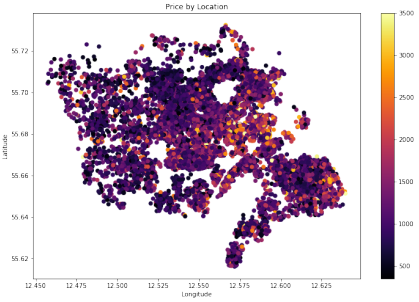
Moreover, given that the data represents real places in a city, it is pertinent to use the 'latitude' and 'longitude' attributes to create some geospatial plots.



(a) Scatter plot of the listings, sorted by neighbourhood



(b) Heatmap of number of listings by location



(c) Scatter plot of the prices by geolocation

These plots provide a straight forward understanding of the areas that have the most listings and the highest prices as if it were on a map. According to the plot, the areas with the most listings are Nørrebro and Vesterbro, while the most expensive are is Indre By. These takeaways are corroborated by the following plots.

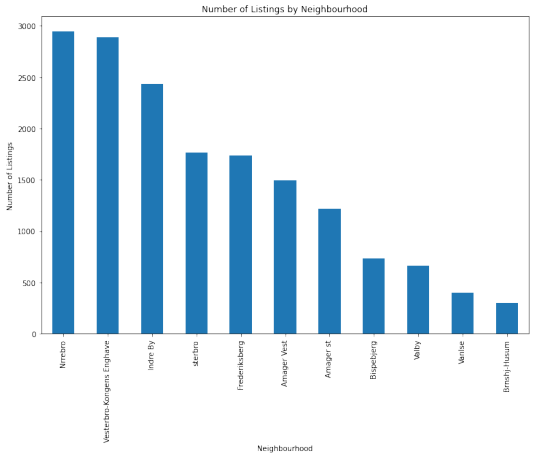
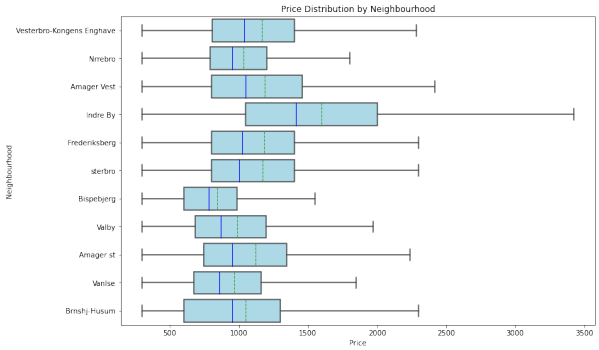
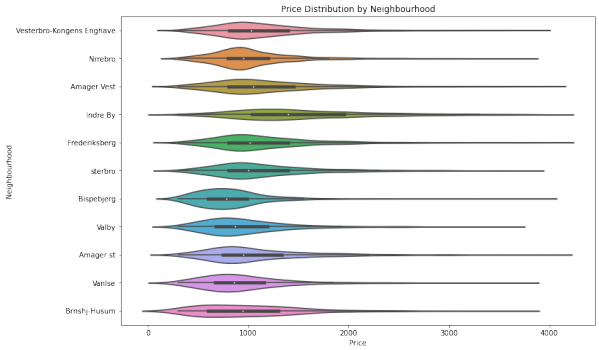


Figure 6: Bar plot of listings by neighbourhood

The plot above is a numeric representation of the same information provided in figure 5b. The plots below are different ways to describe the information in figure 5c in more detail. Each of these plots have their pros and cons, the scatter plot allows the more essential information to be conveyed in a more visual way. The boxplot provides more information, such as the mean, the median and the 25th an 75th percentiles for each neighbourhood, while the violin plot combines both aspects in a way that allows for a different interpretation of the data. Looking the violin plot, it is clear that the price distribution in Norrebro is the most consistent, while Brønshøj-Husum and Indre By have the widest ranges of prices.

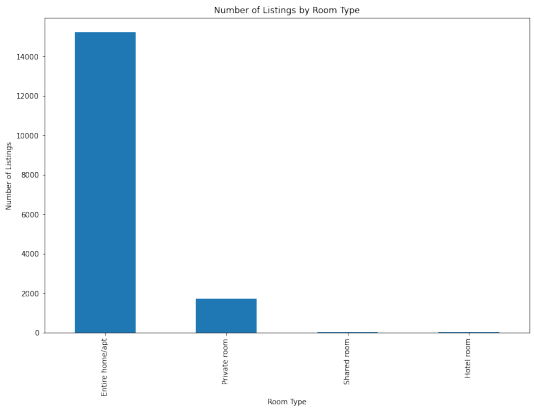


(a) Boxplot of the prices sorted by neighbourhood

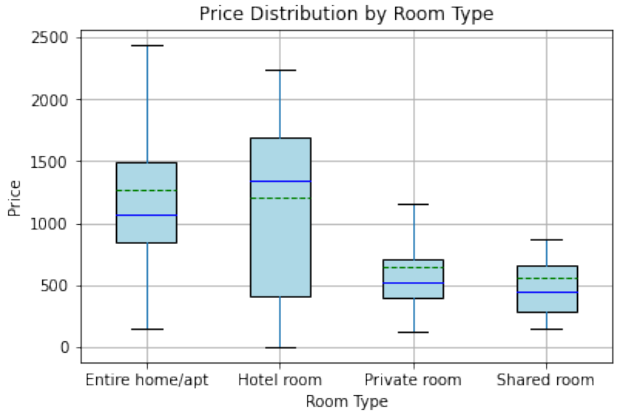


(b) Violin plot of the prices by neighbourhood

Additional plots were created in an attempt to understand how the 'room_type' attribute affects the price. This way, it was verified that the most common type of listing is by far for Entire home/apartment, and that despite the Shared Room and Hotel Room types are a minority, they represent the types with the lowest and highest average prices, respectively.



(a) Number of listings by room type



(b) Price distribution by room type

Unfortunately, however, all these isolated insights provide a very limited understanding of how all the attributes interact and how they affect each other. Even though all these plots can be useful in their own way, a lower-dimensional representation of the high-dimensional dataset is needed, which can be obtained by performing a Principal Component Analysis (PCA).

3.1 Principal Component Analysis

The Principal Component Analysis is a technique sensitive to the scale of our attributes, which means the data must be cleaned and standardized before starting this process. The data has beed previously modified to filter out irrelevant attributes, missing data and outliers. Next, the data attributes will be transformed to be in the appropriate format to perform a PCA.

The features transformation process started with standardizing the 'latitude', 'longitude', 'rating', 'bedroom' and 'bath' attributes. This was done by subtracting the mean and dividing the standard deviation. Next, a One-out-of-K enconding was performed to transform the 'room_type' from a nominal attribute to a number of binary variables, corresponding to the existing types. Finally, the 'minimum_nights' and 'number_of_reviews' attributes were binarized, classified with binary values representing 'high' or 'low'. For this, a specific threshold was set to each attribute, and all the values are swapped by a '0' if they are lower or by '1' if they are greater than the threshold. The median of each attribute was used as the threshold, meaning it is '6' for the 'number_of_reviews' and '3' for the 'minimum_nights' attribute.

At this point, the data is ready to be applied in machine learning models or other techniques such as PCA, and so the dataframe was converted to a matrix of dimension 16584 x 13. Also, after recreating the Pearson Correlation Map with the 'new' data, it's clear that the level of correlation between the attributes has improved. The number of attributes considered 'strongly correlated' has significantly increased, meaning that the transformations had positive results.

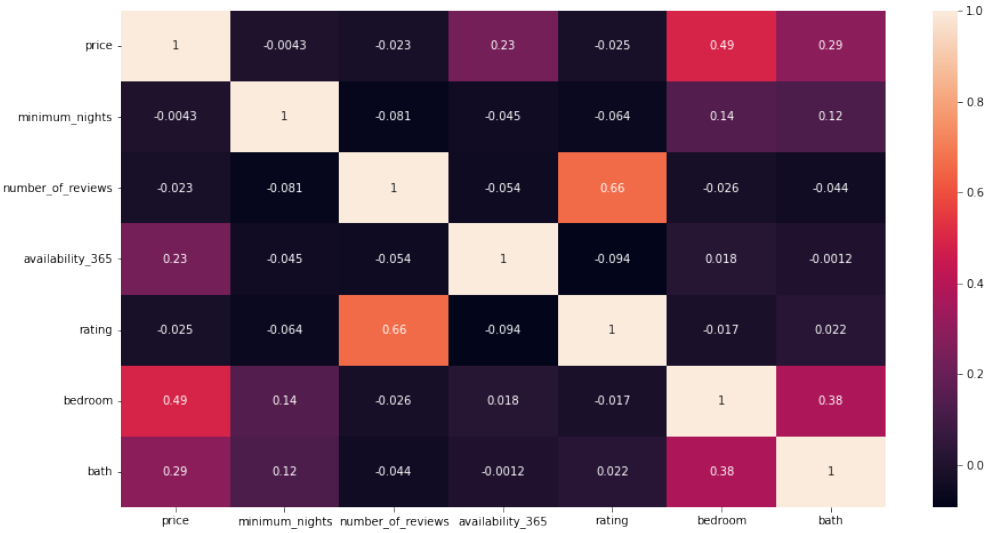


Figure 9: Pearson Correlation Map for clean data

Back to the Principal Component analysis, this is a technique used to reduce the dimension of a data set and extract features. The first step is to subtract the mean of each column and then calculate the Singular Value Decomposition to the matrix. After this, it's time to compute the variance explained by each component by dividing the squared singular value of the given component by the sum of all the squared singular values, which can be plotted for a smoother interpretation.

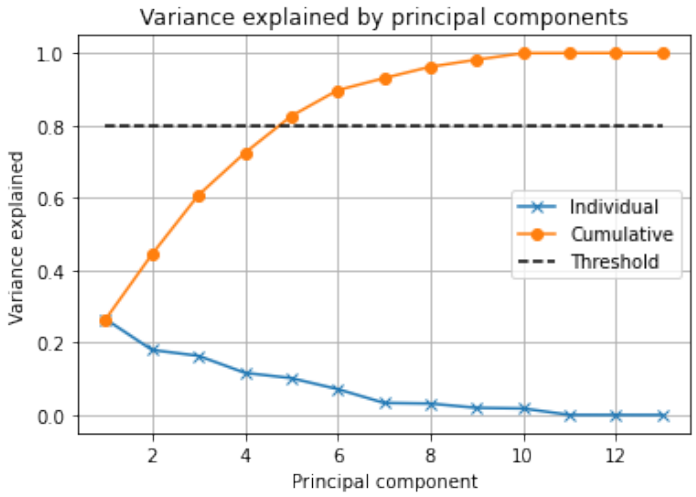


Figure 10: Variance explained and combined variance explained

The takeaway from this plot is that the first 5 components explain a little over 80% of the variance, and as such, the focus will be on the first 5 components from here on out.

The extraction of Eigen vectors during the computing of a PCA produces the coefficients for each component that represent how much each original attribute contributes to the variation captured by the principal component. The following plot allows the analysis of these coefficients for each of the first 5 components in every attribute. If an attribute presents positive coefficients for a component, it means that they are positively associated and that it contributes to the same direction of variation, while negative coefficients is indicative that that attribute contributes to the opposite direction of variation as the component.

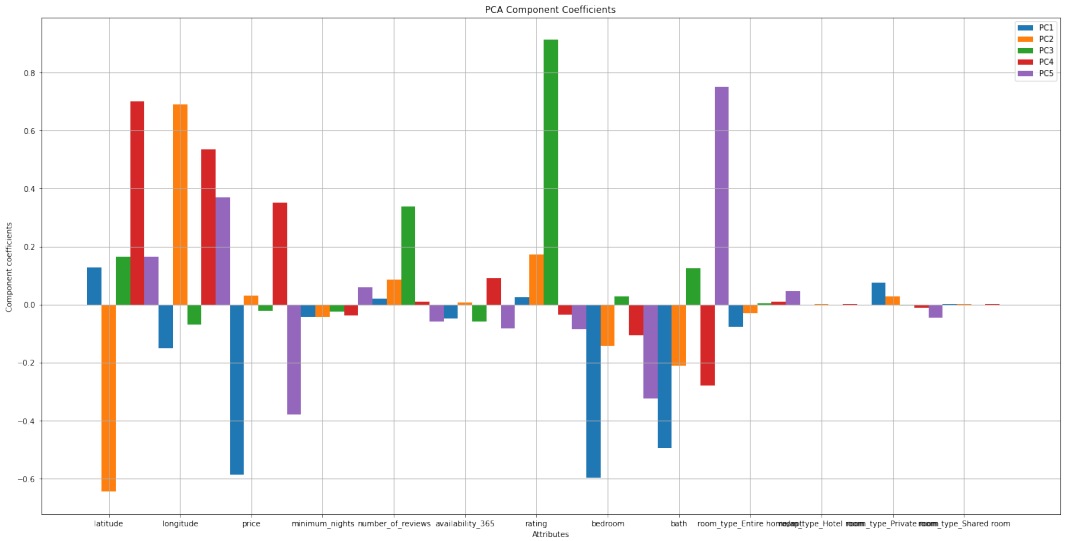


Figure 11: PCA Component Coefficients

Isolated plots of the coefficients for each individual component can be found in section 6. Finally, the PCA was projected to the data, producing the following result.

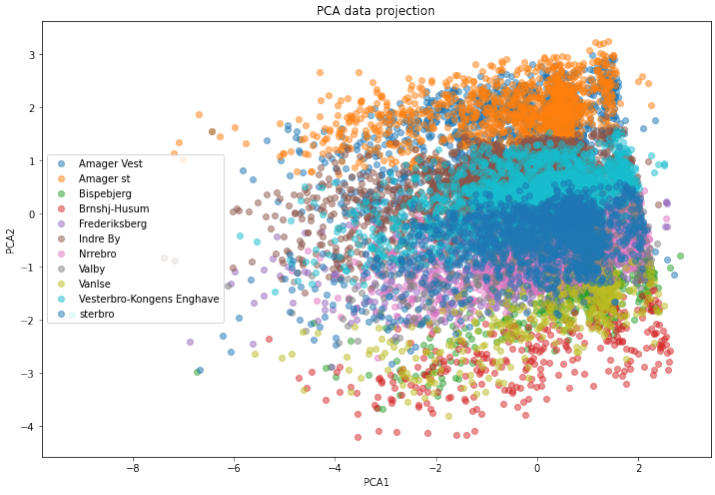


Figure 12: Projection of the PCA onto the data

4 Discussion

In the initial stage of The project, the focus was on understanding the data, selecting the pertinent attributes and analyzing its summary statistics. This led to the removal of irrelevant features, addressing missing data, and mitigating the impact of outliers. Next up, multiple data visualizations were created, which provided a deeper understanding about the dataset in general, as well as the attributes and their relations.

During the process of Principal Component Analysis, it was acknowledged that the attributes needed to be further prepared for data processing. This was done through methods such as standardization, binarization, or thresholding, and one-out-of-K encondig. Feature transformation is important because PCA gives more weight to variables with higher variance, and without it, variables with larger scales would dominate the principal components, and the analysis would be biased towards those variables. These methods allowed for meaningful comparisons to be accessed between the attributes. Variables measured in different units or with different scales would not be directly comparable in a PCA, and so, standardization makes it possible to compare the relative importance of variables in terms of their contributions to the principal components.

In summary, the knowledge attained covers the ability to work with data proficiently, understand the role of the attributes, and implement essential transformations to improve its relevance when constructing the machine learning models.

5 Appendix A

5.1 Exam Problems

Question 1

Answer: **D**

Since 'Time of Day' falls within Calendar dates category, the attribute is **Interval**. Also, 'Traffic lights' and 'Running over' are both counts, which means they are **Ratios**.

Question 2

By applying the following formula

$$\underbrace{\|x\|_p}_{p-Norm} = (\sum_{i}^{n} |x_i|^p)^{\frac{1}{p}} = (|x_1|^p + |x_2|^p + \cdots + |x_n|^p)^{\frac{1}{p}}$$

to calculate the p-norm distance $d_p(x_{14}, x_{18})$, being $x = x_{14} - x_{18} = \begin{bmatrix} 7 \\ 0 \\ 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$,

we found that the highest the p, the closer the p-norm distance was to 7, which is the highest value of x. For this, we can say that $d_{p=\infty}(x_{14}, x_{18}) = 7.0$

Answer: **A**

Question 3

Answer: **A**

$$Var.explained = \frac{\sigma_1^2 + \sigma_2^2 + \sigma_3^2 + \sigma_4^2}{\sigma_1^2 + \sigma_2^2 + \sigma_3^2 + \sigma_4^2 + \sigma_5^2} = \frac{13.9^2 + 12.47^2 + 11.48^2 + 10.03^2}{13.9^2 + 12.47^2 + 11.48^2 + 10.03^2 + 9.45^2} = 0.86 \quad (1)$$

Since 0.86 is greater than 0.8, the correct answer is A.

Question 4

Jaccard's similarity between s1 and s2 can be calculated with the following expression: $J(s1, s2) = \frac{f11}{K - f00}$, where K is the total number of attributes, $f11$ is the number of attributes present is both $s1$ and $s2$ and $f00$ is the number of attributes that are not present in neither $s1$ or $s2$.

$J(s1, s2) = \frac{2}{20000 - 13} = 0.0001$

Answer: **C**

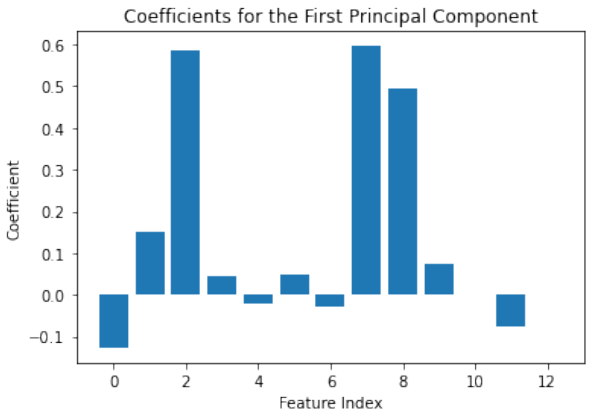
Question 5

Answer: **A**

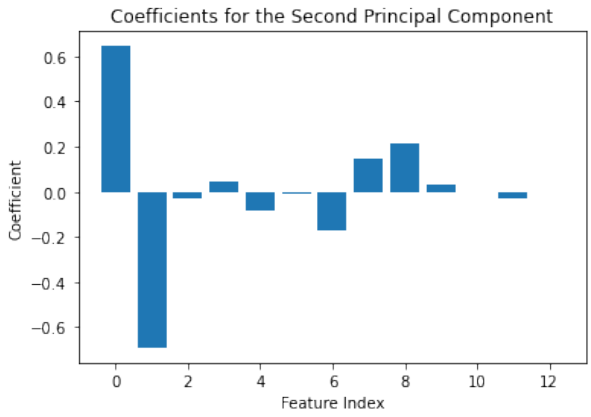
Question 6

Answer: **A**

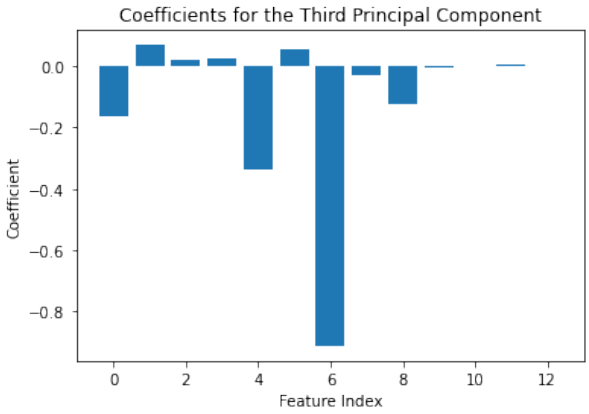
6 Appendix B



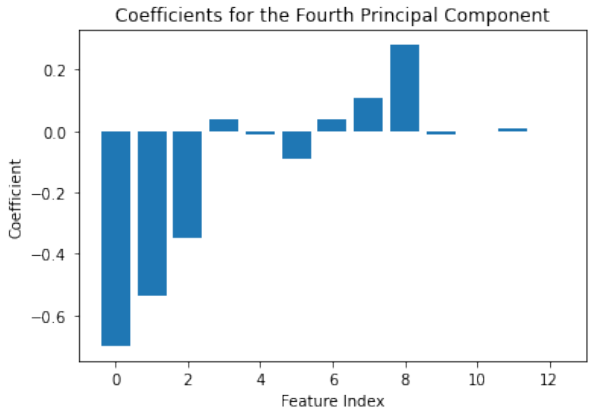
(a) Component 1 Coefficients



(b) Component 2 Coefficients



(a) Component 3 Coefficients



(b) Component 4 Coefficients

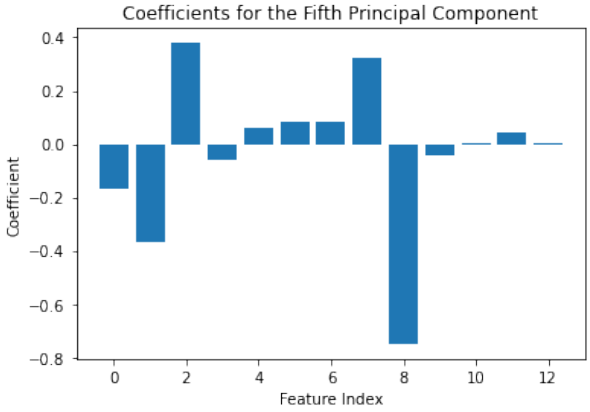


Figure 15: Component 5 Coefficients

References

[1] Jasleen Dhillon, Nandana Priyanka Eluri, Damanpreet Kaur, Aafreen Chhipa, Ashwin Gadupudi, Rajeswari Cherupulli Eravi, and Matin Pirouz. Analysis of airbnb prices using machine learning techniques. In *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0297–0303. IEEE, 2021.

[2] Pouya Rezazadeh Kalehbasti, Liubov Nikolenko, and Hoormazd Rezaei. Airbnb price prediction using machine learning and sentiment analysis. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 173–184. Springer, 2021.