School of CMS  
Cover Sheet for Coursework

Module Code: CO3093

Assignment: Coursework 2

Surname (in CAPITALS): FIROST

First name (in CAPITALS): AKIL MOHAMED

ID Number: 219015127

I understand that this is a piece of coursework. I confirm that this work is mine and that I am fully  
aware of the Academic Integrity declaration within the school of Informatics accessible here.  
Date: 3rd April 2025  
Signature:

A close up of a signature

Description automatically generated

Dataset Overview

The dataset provided is an Airbnb listing in London with the dataset containing 66,679 records and 31 features which contains information about a variety of attributes including host details, property characteristics, availability, reviews, and pricing.

The features that are available within this csv dataset file are as follows :

* **Identification:** id, host\_id, host\_name
* **Textual Metadata:** name, description, amenities
* **Host Information:** host\_since, host\_response\_rate, host\_acceptance\_rate, host\_is\_superhost, host\_listings\_count
* **Location Details:** neighbourhood, latitude, longitude
* **Property Characteristics:** property\_type, room\_type, accommodates, bathrooms, bedrooms, beds
* **Pricing and Policy:** price, minimum\_nights, maximum\_nights
* **Availability & Review History:** calendar\_last\_scraped, number\_of\_reviews, first\_review, last\_review, review\_scores\_rating
* **Platform Metrics:** calculated\_host\_listings\_count

Notably, the price column is stored as a string including a dollar sign and needs to be cleaned for numerical analysis. There are also missing values in critical columns such as review\_scores\_rating, price, bathrooms, and beds, highlighting the need for comprehensive data cleaning. Overall, from my initial observation I could understand that the dataset offers rich and diverse information suitable for predictive analytics, but it requires significant preprocessing to ensure accuracy and consistency in model training.

Data Cleaning

To start preparing for the data for the predictive model, extensive cleaning was required to the data. Since the original dataset contained **66,679 records and 31 columns**, as shown in the initial inspection (**df.shape**). Categorical and numerical features were programmatically separated using **df.select\_dtypes()** to facilitate targeted cleaning. The price column, originally stored as a string with currency symbols (e.g., $200.00), was cleaned using a regular expression and converted to float using:

df['price'] = df['price'].replace('[\\$,]', '', regex=True).astype(float)

The missing values were replaced or handled accordingly, for example empty list entries in columns like **amenities** were converted to **NaN**, and a missing value summary highlighted gaps in key fields such as **description**, **host\_response\_rate**, **bathrooms**, and several review-related columns. But among the many columns there were many irrelevant features such as the **latitude, longitude, and calendar\_last\_scraped** which were dropped as they were of no help / no contribution to the efficiency of prediction model. For those kinds of columns, they were dropped followed by duplicate records and any rows still containing NaN values, reducing the dataset to **41,888 rows and 27 columns**.

After that the next step in the cleaning was the mitigate the impact of extreme pricing outliers, the **interquartile range (IQR)** method was applied. Listings with prices beyond 1.5×IQR from the first and third quartiles were removed using:

df = df[(df['price'] > lower\_bound) & (df['price'] < upper\_bound)]

This further refined the dataset to **39,323 rows**. To address the skewness in price distribution, a natural logarithm transformation was applied:

(df['log\_price'] = np.log1p(df['price']))

Producing a more normally distributed target variable. Lastly, all numerical columns were standardised using **StandardScaler()** to ensure equal weighting during clustering and regression.

Data Exploration

To understand the patterns within the dataset and help build the features for the model a series of exploratory data analyses (EDA) were conducted. These visualisations focused on pricing distributions, correlations, and relationships between key features and the target variable (log\_price). Overall, the analysis revealed strong spatial and structural patterns in pricing and helped to isolate the most relevant predictors for modelling.

A graph of blue rectangular bars with white text

Description automatically generated

The first visualization is mainly focussed on the neighbourhood-based pricing patterns. For the ease of understanding the top 15 neighbourhoods were chosen with the visualization being a bar chart format that were based on the average of the log-transformed listing prices. The visualisation clearly shows that the **City of London**, **Kensington and Chelsea**, and **Westminster** consistently lead in terms of average price.

Based on the current geographic map evidence these places hold significant importance such as the historic importance, central location, etc. On the other hand, places such Newham and Greenwich showed comparatively lower average prices, suggesting that properties in these areas are more budget-friendly or less in demand.

A graph with a line going up

Description automatically generated

The next visualization explores the how the guest capacity /accommodation influences the pricing. Upon interpreting what I could understand was that there was strong relation between the number of guests and the average pricing, especially when staring from the number of guests being between 1 to 4.

For example, listings that accommodate 1 person have an average log price around **4.00**, while those that accommodate 4 guests have an average log price exceeding **5.60**, indicating nearly **a 1.6-fold increase** in raw price terms (since log(Price) ≈ 5.6 corresponds to ~270 GBP vs ~55 GBP for log(Price) ≈ 4.0).

But once we reach the 4 number of guests, we can see that the line dips suggesting that adding more guest capacity does not necessarily need to increase price. This could be due to market price ceilings, reduced demand for large group listings, or limited premium features beyond a certain size. Once again, we can once again see that the line has slightly increased dip between guests 5 to 6 meaning that this is likely due to fewer listings in these capacity brackets, which introduces variance and limits the reliability of the average.

A screenshot of a graph

Description automatically generatedThe next visualization is about the understanding of between the relationship of multiple numerical features and for that the scatter matrix was used and in this matrix the variables most relevant to pricing:

log\_price, accommodates, bedrooms, minimum\_nights, number\_of\_reviews, and review\_scores\_rating, with each of the subplot showing the scatterplot between two variables and the diagonal representing **kernel density plots (KDEs)** of each variable’s distribution. Using this plot, we can understand and identifying linear or non-linear trends, clustering patterns, skewed data, or outliers that may influence model performance.

Some of the key understandings were as follows :

* **Log price vs accommodates,** and **log price vs bedrooms** show moderate linear relationships. Listings with more rooms and higher guest capacity tend to be priced higher. This validates the inclusion of both variables as predictors in later regression models.
* **Minimum nights**, **number of reviews**, and **review\_scores\_rating** do **not** show a strong relationship with log\_price, suggesting their influence is weaker. These plots appear scattered with no clear pattern.
* **Accommodates vs bedrooms** displays a strong positive linear trend, which is logical — larger guest capacity typically requires more rooms. However, this also indicates potential **multicollinearity** between these two predictors, which could be problematic in linear models unless addressed via regularization or clustering.
* Many variables (e.g., minimum\_nights, number\_of\_reviews) show **right-skewed distributions** — long tails and many low values — justifying earlier **log transformations and standardization** applied during preprocessing.

A screenshot of a graph

Description automatically generatedIn the correlation matrix, we are using the heatmap for this as the matrix helps inform **feature selection decisions**, ensuring that only meaningful predictors are retained while highly correlated ones are monitored for multicollinearity. It also provides numerical validation to the earlier visual trends identified in scatter plots and supports the case for log transformation (normalization of skewed variables) performed in earlier preprocessing stages. By making use of the colour coding and the Pearson correlation coefficient between key numerical features after preprocessing help in identifying the strong linear relationship. In this case the coefficient ranges from -1 to 1 with the implications as follows :

* **1** implies a perfect positive linear relationship,
* **-1** implies a perfect negative relationship,
* **0** indicates no linear relationship.

With the colour coding representing as follows, warmer tones (reds) signifying stronger positive correlations and cooler tones (blues) indicating weaker or negative relationships.

The key understanding from this heat map was the following :

* The **log-transformed price (log\_price)** shows a **moderate positive correlation** with:
  + accommodates (0.62)
  + bedrooms (0.47)  
    This validates the exploratory findings and supports the inclusion of these features in the regression model.
* The correlation between **accommodates and bedrooms** is very strong (0.76), indicating potential **multicollinearity**.
* Other features such as minimum\_nights, number\_of\_reviews, and review\_scores\_rating exhibit **very weak or no correlation** with log\_price (correlation coefficients close to 0).

A diagram of a graph

Description automatically generatedSince we have explored how the price vary based on the number of guests we also need to know how the price would vary based in the room type and for that reason by making use boxplot would help us understand how much the guests are likely to pay based on the level of privacy and space offered.

Each box represents the **interquartile range (IQR)** of log prices for that room type, with the **line inside the box indicating the median**. The **whiskers extend to 1.5 times the IQR**, and any points outside this range are considered **outliers**.

This boxplot is valuable for comparing **central tendency** (median), **spread** (IQR), and **outlier behaviour** across categories.

The key understanding from this boxplot were the following :

* **Hotel rooms** and **Entire home/apartment** listings have the **highest median log prices**, suggesting that privacy and space command premium pricing.
* **Private rooms** and especially **Shared rooms** exhibit **lower median prices**, indicating affordability but with significantly more variation.
* **Hotel rooms** show a relatively tight spread, indicating price consistency, possibly due to standardised pricing across similar listings.
* **Private rooms** display a **wider range** of prices and a high number of outliers on the upper end, suggesting variability in quality, location, or amenities.
* A few outliers in **Shared rooms** and **Private rooms** approach the same price level as Entire homes, possibly indicating misclassifications or exceptionally high-end rooms.

Model Building

In this phase of the building the model, where the main aim is to predict Airbnb listing prices in London I used **linear regression** approach. Instead of predicting the raw price which more of right skewed due to the outliers, we instead use the log\_price for better model performance and distributional properties. To proceed the next step was to choose which feature was to be selected for the model training and based on the EDA and correlation matrix findings, the following features were selected for model training:

* **Numerical Features**: accommodates, bedrooms
* **Categorical Features**: room\_type, neighbourhood

A screen shot of a computer

Description automatically generatedAnd after that the data was then split into 80% data training and 20% testing with ‘x’ being for features and ‘y’ being for target variable (log\_price).

Once the data split and the assigning how much to training and test was done the next step was train the actual model by making use of the standard **Linear Regression** model from **sklearn.linear\_model** was used. It was trained using the training set (**X\_train, y\_train**) and then applied to predict on the testing set (**X\_test**).

A screen shot of a computer program

Description automatically generated

Finally, after the model was done training to evaluate how accurate the model prediction was, a model evaluation was run where the 3 key regression metrics were run and on top of that a 5-fold cross validation was also run in order to understand how the model generalizes. The following were results :

|  |  |  |
| --- | --- | --- |
| **Metric** | Value | Interpretation |
| R² Score | 0.696 | ~70% of the variance in log price is explained by the model — a good fit for linear regression |
| MAE (Mean Absolute Error) | 0.281 | On average, predictions are within 0.28 units of log price from the actual |
| RMSE (Root Mean Squared Error) | 0.359 | The model’s typical error magnitude, penalizing larger errors more heavily |

And based on the **5-fold cross-validation,** the following were the results (the results have been rounded up) :

**Cross-validated R² scores**: [0.684, 0.708, 0.717, 0.676, 0.664]

**Average CV R²**: **0.690**

This indicates that the values are close to the main test-set R² (0.696), indicating low overfitting and good generalisation performance.

Improved model

Following the linear regression model, to enhance the performance of the previous model a K-Means clustering approach was used. By making use of the k-means clustering and using the local regressor on the clusters, the aim of this step is to capture heterogeneity in the dataset by identifying the clusters with similar listing characteristics. Moving forward just like when budling the model once again we have to choose and specify which are the features that will be used during this model building and my choosing criteria was the features has to represent both the capacity and quality of the listings, so accordingly these were the features chosen accommodates, bedrooms, minimum\_nights, number\_of\_reviews, and review\_scores\_rating. Also, before the clustering starts, the features were standardized using StandardScaler to ensure that all variables contributed equally to the clustering algorithm, in other words we had to normalize the features prior to clustering.

A graph with a line

Description automatically generatedAfter the normalization the next step in the process was to obtain the optimal number of Cluster (K) using the elbow method.

By using the elbow method to determine the optimal number for K. First, we plot the sum of squared error (SSE) for the y axis and K ranging from 1 to 10 for the x axis. In the plot we identify that the elbow pattern or the bottom point of the elbow indicated the optimal number for K. In our case our optimal number for K will 4. When K is more than the cluster will not result in any significant improved modelling of the data.

A diagram of a cluster of dots

Description automatically generatedNow that we have found the optimal number for K the next we need to do is to validate the cluster visually, this means that we need to plot them and for that I used Principal Component Analysis (PCA) to reduce the high-dimensional feature space to two principal components. The scatterplot in the figure shows that clusters are well sperated in the 2D space and cluster 3 appears to be isolated, meanging that it is representing outlier listings such as luxury properties or very niche accommodations. This separation confirms that K-Means was successful in capturing meaningful structure in the data.

Final segement of this improved model was to do local regressor on each of the clusters and for each of them separate linear regression model was trained using the same features as the global model: accommodates, bedrooms, room\_type, and neighbourhood. The follwing table will help you understand the performance in each of the clusters :

|  |  |  |  |
| --- | --- | --- | --- |
| Cluster | R² Score | MAE | RMSE |
| 0 | 0.622772 | 0.279183 | 0.360331 |
| 1 | 0.337979 | 0.258740 | 0.332905 |
| 2 | 0.053591 | 0.375231 | 0.540239 |
| 3 | 0.651618 | 0.279676 | 0.354416 |

Following the table what we can understand is that :

* Cluster 3 achieved the highest R² Score = 0.652, this means that it is outperforming even the original global model (0.696).
* Cluster 1 showed reasonable error metrics but a lower R², point to the fact that limited variance explained despite good prediction consistency.
* Cluster 2 had the weakest performance, likely due to representing outlier or highly diverse listings, which are difficult for linear models to handle.

Conclusion

To conclude my main objective during the time of work was to predict the price of Airbnb London listing using predictive model. By carrying out tasks such as data cleaning, handling missing data, treating outliers, doing log-transforming the price variable and then doing exploring the clean the dataset through visualizations, I was able to narrow down and identify the key features that were essential to building the model and the features being accommodates, bedrooms, room\_type, and neighbourhood.

Then moving forward to start the build of the model, I used linear regression model which was trained using selected numerical and categorical features. This gave pretty good result based on the evaluation metric, with the R² Score being 0.696 and consistent cross-validation scores that indicated good generalisation. But to further improve the model, K-means clustering technique was used. This allowed me to split the data into four distinct clusters that were based on similar property characteristics, and train individual linear regression models within each cluster. The main idea behind this was that a local model within a more homogeneous group can often predict better than a single global model and from the result I could see that some clusters performed better than the global model, with a high R² and low error values, while others like Cluster 2 struggled. As a result, the enhanced model demonstrates that although clustering can enhance performance in certain domains, it might not necessarily yield superior outcomes in every section.

All things considered, this project improved my understanding of how to carefully apply model improvement techniques like clustering and how to develop a predictive model from raw data to final evaluation.