airbnb

# Group Report

# Unsupervised ML for Airbnb


**Group no.** 8

**Names:** Anna Meudec, Chloe Downes, Eris Byrne, Kellie Staunton, Maxime Junca Quintero.

**Module Code:** MT413

**Lecturer:** Mathieu Mercadier

# Contents

**DCU University's Declaration on Plagiarism Assignment Submission Form**

This form **must** be filled in and completed by ***all the students*** submitting an assignment. Assignments submitted without the completed form will not be accepted.

**Names:** Anna Meudec, Chloe Downes, Eris Byrne, Kellie Staunton, Maxime Junca Quintero.

**Programmes:** INTB4, BSI4, EBF4, BS3
**Module Code:** MT413
**Assignment Title:** Classical Unsupervised Learning - Group Report
**Submission Date:** 27/11/2023

We declare that this material, which we now submit for assessment, is entirely our own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. We understand that plagiarism, collusion, and copying are grave and serious offences in the university and we accept the penalties that would be imposed should we engage in plagiarism, collusion or copying. We have read and understood the Assignment Regulations set out in the module documentation. We have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the sources cited are identified in the assignment references. This assignment, or any part of it, has not been previously submitted by members of the group or any other person for assessment on this or any other course of study.

We have read and understood the referencing guidelines found at https://www101.dcu.ie/library/Citing&ReferencingGuide/player.html and/or recommended in the assignment guidelines.

| **Names Signed:** | | **Date:** |
|---|---|---|
| *Anna Meudec* | *(20426856)* | *25th November 2023* |
| *Chloe Downes* | *(21106657)* | *25th November 2023* |
| *Eris Byrne* | *(20369023)* | *25th November 2023* |
| *Kellie Staunton* | *(20741531)* | *25th November 2023* |
| *Maxime Junca Quintero* | *(22107517)* | *25th November 2023* |

## Introduction

With the travel industry expected to grow, on average, 5.8 per cent annually through to 2032 (WTTC, 2022), travel organisations, now more than ever, must embrace the implementation of new and existing technologies to unlock sustainable long-term growth (Almasi et al., 2023). One such organisation, nested between the hospitality and travel industry, that has continually disrupted the conventional reservation system is Airbnb. Airbnb, a highly popular P2P hosting platform and a frontrunner in booking and leasing tourist accommodations, will become our focus organisation for this report (Zhu et al., 2019).

One prominent issue faced by Airbnb's hosts and highlighted by its vice president is that "it's pretty hard for them to know how to price their listings" (Venture Beat, 2017) and, generally, "solely depends on the host to set their own prices" (Dhillon et al., 2021, p. 0297). With this, our report aims to use classical unsupervised machine learning to identify patterns in Airbnb's property listings to assist hosts in setting competitive listing prices. Machine learning (ML), deemed a "crucial branch of artificial intelligence" (Bamisaye & Alabi, 2023, p. 121), can be found across all areas of our daily lives and plays a pivotal role in almost all industries, including travel (Bulanov, 2023). The integration of ML has transformed the travel industry by leveraging data insights to better understand consumers and their behaviours and make effective data-driven decisions. With this, travel organisations have additionally been able to significantly tailor their products and offerings to their consumers, which, in turn, can reduce costs, improve overall performance and aid in delighting consumers (Swetaseal, 2023).

Our chosen method, classical unsupervised learning, is a subset of ML in which algorithms "identify common elements and recognise useful structures and patterns from input data without requiring the data to be labelled" (Egger, 2022, p. 91). Through identifying hidden patterns using unsupervised learning, Airbnb hosts can optimise the likelihood of receiving bookings and mitigate potential revenue loss due to under or overpricing their properties. In addition, hosts will be able to see the underlying relationship that exists between property features and average price. Thus, they can make pricing decisions and potentially invest in new features to increase revenue or achieve a competitive advantage on Airbnb. Ultimately, this approach aims to successfully bring more business to both Airbnb and its hosts, increase revenues, and match consumers' accommodation preferences faster and easier while ensuring a marketplace of reasonably priced properties.

Our overall proposal in this report involves the analysis of an Airbnb dataset of different regions in the United States. Firstly, this report will explore the appropriate data cleaning and feature engineering methods, creating a solid backbone to ensure an accurate and reliable model. Our chosen model will then be outlined and its quality validated, followed by our findings and results. Moreover, we will propose specific software and tools designed for seamless integration into Airbnb's operations, accompanied by acknowledgements of any identified limitations. Conclusively, we will summarise our key findings and present actionable recommendations whilst suggesting potential avenues for future research.

## Data Cleaning and Feature Engineering

This section will outline the methodologies employed in the data cleaning and feature engineering of our dataset on Airbnb listings. The purpose of this step in the process is to address errors, anticipate challenges and enhance features to prepare the data for building our unsupervised learning model. Walker (2022) reminds us that this part of the process should not be seen as an individual step but rather as a fundamental element persistently intertwined. Therefore, before cleaning the data, it is essential to understand the dataset's features to inform decision-making when handling missing/non-values.

## Data Cleaning

**Handling Missing Values:**
In the context of travel data, missing values might arise due to incomplete bookings, unreported preferences, or system errors. In the case of our specific dataset on property listings, missing data may also occur due to duplicated or incomplete listings. Techniques such as imputation, where missing values are filled based on statistical measures or algorithms, can help maintain data integrity. However, filling the missing data with measures such as the mean, median, and mode can change the results, skewing the results or creating incorrect biases. To handle missing values, we used descriptive statistics from the Pandas library to understand the individual features better.

Fig. 1. Describing the Data

```
BnB_Df.describe(include="all")
```

| | name | host_id | host_name | neighbourhood_group | neighbourhood | la |
|---|---|---|---|---|---|---|
| **count** | 232131 | 2.321470e+05 | 232134 | 96500 | 232147 | 232147.0 |
| **unique** | 220136 | NaN | 29367 | 30 | 1412 | |
| **top** | Presidential Suite In A Mansion | NaN | Blueground | City of Los Angeles | Unincorporated Areas | |
| **freq** | 150 | NaN | 4305 | 22204 | 11882 | |
| **mean** | NaN | 1.582248e+08 | NaN | NaN | NaN | 36.6 |
| **std** | NaN | 1.587164e+08 | NaN | NaN | NaN | 5.1 |
| **min** | NaN | 2.300000e+01 | NaN | NaN | NaN | 25.9 |
| **25%** | NaN | 2.299242e+07 | NaN | NaN | NaN | 33.9 |
| **50%** | NaN | 1.005783e+08 | NaN | NaN | NaN | 36.1 |
| **75%** | NaN | 2.686930e+08 | NaN | NaN | NaN | 40.7 |
| **max** | NaN | 5.069384e+08 | NaN | NaN | NaN | 47.7 |

After investigating each feature, we displayed the percentage of missing values in each column to decide how best to handle them. As seen below, the columns with high null values were (a) “neighbourhood_group”, (b)“last_review” and (c) “reviews_per_month”. (a) was handled through imputation, as we could easily replace the missing values with “not specified and still hold the value of the data in the remaining columns. (b) and (c) were handled by deleting the rows containing null values in these columns, as the data from these listings would likely not add much value. Listings without reviews likely have not been booked, the listing is incomplete, or simply unavailable to book.

Fig. 2. Fill Na

```
BnB_Df["neighbourhood_group"].fillna(value="Not Specified", inplace=True)
(BnB_Df.isna().sum(axis=0)/len(BnB_Df))

name                              0.000069
host_id                           0.000000
host_name                         0.000056
neighbourhood_group               0.000000
neighbourhood                     0.000000
latitude                          0.000000
longitude                         0.000000
room_type                         0.000000
price                             0.000000
minimum_nights                    0.000000
number_of_reviews                 0.000000
last_review                       0.211439
reviews_per_month                 0.211439
calculated_host_listings_count    0.000000
availability_365                  0.000000
number_of_reviews_ltm             0.000000
city                              0.000000
dtype: float64
```

Fig. 3. Drop Na

```
BnB_Df.dropna(subset=["reviews_per_month","last_review","name","host_name"],
BnB_Df
```

| id | name | host_id | host_name | neighbourhood_group | neighbourhood | l: |
|---|---|---|---|---|---|---|
| 9.580000e+02 | Bright, Modern Garden Unit - 1BR/1BTH | 1169.0 | Holly | Not Specified | Western Addition | 37.7 |
| 5.858000e+03 | Creative Sanctuary | 8904.0 | Philip And Tania | Not Specified | Bernal Heights | 37.7 |
| 8.142000e+03 | Friendly Room Apt. Style - UCSF/USF - San Franc... | 21994.0 | Aaron | Not Specified | Haight Ashbury | 37.7 |
| 8.339000e+03 | Historic Alamo Square Victorian | 24215.0 | Rosy | Not Specified | Western Addition | 37.7 |

Fig. 4. Missing Values

```
#Handling missing / Nan values
(BnB_Df.isna().sum(axis=0))
print(BnB_Df.isna().sum(axis=0)/len(BnB_Df))

name                              0.000069
host_id                           0.000000
host_name                         0.000056
neighbourhood_group               0.584315
neighbourhood                     0.000000
latitude                          0.000000
longitude                         0.000000
room_type                         0.000000
price                             0.000000
minimum_nights                    0.000000
number_of_reviews                 0.000000
last_review                       0.211439
reviews_per_month                 0.211439
calculated_host_listings_count    0.000000
availability_365                  0.000000
number_of_reviews_ltm             0.000000
city                              0.000000
dtype: float64
```

**Standardising Formats:**
Data from various sources often come in different formats. For instance, date formats, currency representations, and location notations may differ. Standardising these formats ensures consistency and facilitates a more straightforward analysis. For our chosen dataset, we converted the "neighbourhood_group" feature. We converted the column to a string before importing the CSV file as there were inconsistencies within the column, which would have caused errors further in the process.

Fig. 5. Importing & Formatting Data

```
Inpath= "C:/Users/kelli/Documents/DCU/Y4/Mt413-Data Mining/Group Assignment [
column_types = {'neighbourhood_group': str}
BnB_Df=pd.read_csv(Inpath+"US_BNB_2023.csv", delimiter=","
                   ,header=0, index_col=0, dtype = column_types)
BnB_Df
```

**Removing Duplicates and Outliers:**
Duplicate entries and outliers can distort the analysis. For example, duplicate booking records might skew demand forecasting. Robust data cleaning practices identify and handle such discrepancies.

Fig. 6. Removing Duplicate Rows

```
#Checking for duplicated rows, deleting them while keeping the first instance.
print("There are " + str(BnB_Df.duplicated().sum()) + " duplicated Rows")

#return duplicated rows
duplicated_rows = BnB_Df[BnB_Df.duplicated(keep=False)]
print(duplicated_rows)

#remove duplicated rows.
BnB_Df_unique=BnB_Df.drop_duplicates(keep="first")
BnB_Df_unique
```

**Data Smoothing:**

Data smoothing methods like moving averages or filters are employed to reduce noise or irregularities in the dataset, aiding in identifying trends or patterns. There are many ways to identify noise in data, including visualisation and statistical analysis. For our dataset on Airbnb Listings, we statistically analysed the numerical and categorical data to identify noise.

Fig. 7. Investigating Data

```
#Investigating categorical data.
BnB_Df_unique.describe(include="object")
```

| | name | host_name | neighbourhood_group | neighbourhood | room_type | last_review | city |
|---|---|---|---|---|---|---|---|
| count | 183043 | 183043 | 183043 | 183043 | 183043 | 183043 | 183043 |
| unique | 178445 | 25998 | 31 | 1406 | 4 | 3147 | 27 |
| top | Grand Desert Resort - 2 Bedroom Deluxe | David | Not Specified | Unincorporated Areas | Entire home/apt | 05/03/2023 | New York City |
| freq | 52 | 1393 | 110446 | 8743 | 137271 | 5008 | 32618 |

```
#Investigating Numerical data.
BnB_Df_unique.describe(include=np.number)
```

| | host_id | latitude | longitude | price | minimum_nights | number_of_reviews | reviews_per_mon |
|---|---|---|---|---|---|---|---|
| count | 1.830430e+05 | 183043.000000 | 183043.000000 | 183043.000000 | 183043.000000 | 183043.000000 | 183043.0000 |
| mean | 1.461691e+08 | 36.580045 | -98.160240 | 225.667543 | 11.296482 | 51.891026 | 1.6384 |
| std | 1.520465e+08 | 5.207822 | 19.599854 | 906.075898 | 25.676591 | 87.633482 | 1.9108 |
| min | 2.300000e+01 | 25.957323 | -123.088120 | 0.000000 | 1.000000 | 1.000000 | 0.0100 |
| 25% | 2.059506e+07 | 33.893775 | -118.288105 | 90.000000 | 1.000000 | 4.000000 | 0.3100 |
| 50% | 8.333000e+07 | 36.194640 | -97.716010 | 145.000000 | 2.000000 | 18.000000 | 1.0000 |
| 75% | 2.445638e+08 | 40.715150 | -77.060225 | 239.000000 | 29.000000 | 60.000000 | 2.4200 |
| max | 5.059515e+08 | 47.734010 | -70.996000 | 100000.000000 | 1250.000000 | 3091.000000 | 101.4200 |

See below some examples of where we identified noise and how it was handled.

1. For “minimum_nights,” the max stay was significantly higher than the 75th percentile. We investigated to understand that outliers were skewing it.

Fig. 8. Handling Noise (a)

```
# Looking at the difference between the Mean, 75th percentile and the Max values of the "minimum_ni
#We identified that we should remove values above 365 (the number of days in a year).
BnB_Df_unique= BnB_Df_unique[BnB_Df_unique["minimum_nights"]<=365]
BnB_Df_unique.describe(include=np.number)
```

| | host_id | latitude | longitude | price | minimum_nights | number_of_reviews | reviews_per_mon |
|---|---|---|---|---|---|---|---|
| count | 1.829740e+05 | 182974.000000 | 182974.000000 | 182974.000000 | 182974.000000 | 182974.000000 | 182974.0000 |
| mean | 1.461812e+08 | 36.580171 | -98.159832 | 225.613852 | 11.004766 | 51.902341 | 1.6389 |
| std | 1.520508e+08 | 5.208151 | 19.600210 | 905.946466 | 20.095397 | 87.646000 | 1.9110 |
| min | 2.300000e+01 | 25.957323 | -123.088120 | 0.000000 | 1.000000 | 1.000000 | 0.0100 |
| 25% | 2.060991e+07 | 33.893411 | -118.288283 | 90.000000 | 1.000000 | 4.000000 | 0.3100 |
| 50% | 8.333000e+07 | 36.194680 | -97.715974 | 145.000000 | 2.000000 | 18.000000 | 1.0000 |
| 75% | 2.445638e+08 | 40.715150 | -77.060130 | 239.000000 | 29.000000 | 60.000000 | 2.4200 |
| max | 5.059515e+08 | 47.734010 | -70.996000 | 100000.000000 | 365.000000 | 3091.000000 | 101.4200 |

2. For "reviews_per_month," the max value seemed high as there are only 30 days in a month and a 14-day window for completing a review.

Fig. 9. Handling Noise (b)

```
# Investigating "reviews_per_month" feature. The max value seemed high as there are only
#30 days in a month and there is a 14 day window for completing a review.

#Examining through percentiles.
percentiles = [25, 50, 75, 90, 95]

percentiles.extend([i/10 for i in range(990, 1000, 1)] + [100])

# Calculate the percentiles of the 'price' column
price_percentiles = np.percentile(BnB_Df['reviews_per_month'], percentiles)

# Print the calculated percentiles
for percentile, value in zip(percentiles, price_percentiles):
    print(f'{percentile}th Percentile: {value:.2f}')
```

```
25th Percentile: 0.31
50th Percentile: 1.00
75th Percentile: 2.42
90th Percentile: 3.97
95th Percentile: 5.02
99.0th Percentile: 7.71
99.1th Percentile: 7.89
99.2th Percentile: 8.09
99.3th Percentile: 8.34
99.4th Percentile: 8.67
99.5th Percentile: 9.00
99.6th Percentile: 9.42
99.7th Percentile: 10.04
99.8th Percentile: 10.92
99.9th Percentile: 13.22
100th Percentile: 101.42
```

```
#It is clear from the percentiles that 101.42 is an outlier so we will
#keep values less than or equal to 13.22
BnB_Df_unique= BnB_Df_unique[BnB_Df_unique["reviews_per_month"]<=13.23]
BnB_Df_unique.describe(include=np.number)
```

After we smoothed out the data, we created a subset of the data with the cleaned data, keeping only the columns we deemed valuable. To reduce the computing power required when working with the set going forward.

Fig. 10. Creating Subset

```
BnB_Df_unique_numeric=BnB_Df_unique[["latitude","longitude","price","minimum_nights"
                                     ,"number_of_reviews","reviews_per_month"
                                     ,"availability_365","number_of_reviews_ltm"]]
BnB_Df_unique_numeric
```

**Data Normalisation:**

Data normalisation techniques such as Min-Max scaling or Z-score normalisation are used to standardise the range of values within different variables. These processes are integral in enhancing the data's quality, consistency, and usability for subsequent analysis and modelling within the tourism and travel industry. To determine the best method to normalise our data, we used the skew and kurtosis features of the SciPy Library to understand the general distribution of the dataset.

Fig. 11. Investigating Distribution

```
# Investigate if the dataset is Normally distributed in order to choose a method.

#Calculate skewness for all columns
skewness = BnB_Df_unique_numeric.skew()
print("Skewness:")
print(skewness)

# Calculate kurtosis for all columns
kurtosis = BnB_Df_unique_numeric.kurtosis()
print("\nKurtosis:")
print(kurtosis)
```

As the data was not normally distributed, we utilised the IQR method to normalise the data.

Fig. 12. Normalising the Data

```
#Identify the upper and lower bounds using the IQR method for the 'price' variable
Q1=BnB_Df_unique_numeric["price"].quantile(.25)
Q3=BnB_Df_unique_numeric["price"].quantile(.75)
print("Q1=", Q1)
print("Q3= ", Q3)

IQR=Q3-Q1
print("IQR= ", IQR)

Lwr_bound=Q1-1.5*IQR
Upr_bound=Q3+1.5*IQR
print("Lower Bound= ",Lwr_bound)
print("Upper Bound= ",Upr_bound)

IQR_Out_BnB_Df_unique_numeric=BnB_Df_unique_numeric[(BnB_Df_unique_numeric["price"]>Lwr_bound) &
                                                    (BnB_Df_unique_numeric["price"]<Upr_bound)]
IQR_Out_BnB_Df_unique_numeric
```

```
Q1= 90.0
Q3=  239.0
IQR=  149.0
Lower Bound=  -133.5
Upper Bound=  462.5
```

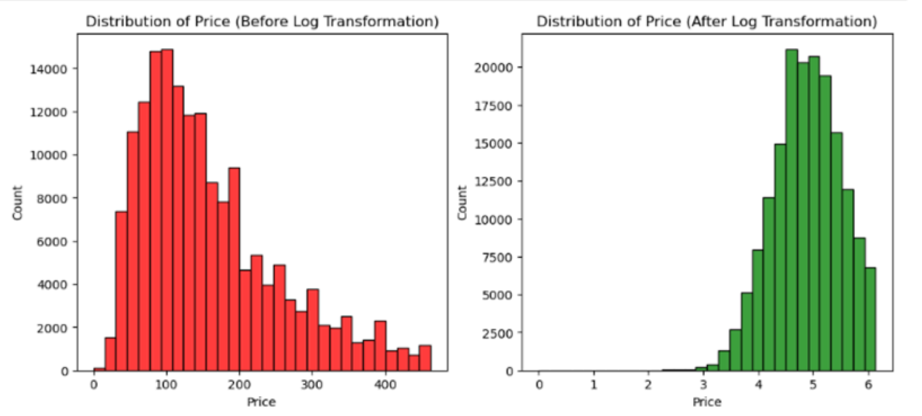
```
#As the majority of the data is not normally distributed and our
#data contains extreme outliers in the price column we will use the IQR to smooth out the data.
```

```
# Smooth out the large values in noOutlrIqrDf using the Logarithm (np.log) and display the min and
IQR_Out_BnB_Df_unique_numeric_log = IQR_Out_BnB_Df_unique_numeric.apply(np.log)
IQR_Out_BnB_Df_unique_numeric_log
```

We also used visualisation techniques to display the dataset before and after normalisation. See below.

Fig. 13. Visualising the Distribution

```
# Displaying the distribution of the "price" after smoothing using the Log transformation.
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
sns.histplot(IQR_Out_BnB_Df_unique_numeric['price'], color="r", bins=30)
plt.title('Distribution of Price (Before Log Transformation)')
plt.xlabel("Price")
plt.subplot(1, 2, 2)
sns.histplot(IQR_Out_BnB_Df_unique_numeric_log['price'], color="g", bins=30)
plt.title('Distribution of Price (After Log Transformation)')
plt.xlabel("Price")
plt.show()
```



## Feature Engineering

### Creating New Features:

Feature Engineering involves creating new features from existing ones to enhance the model through utilising domain knowledge and data understanding to engineer meaningful features that could provide valuable insights to the model. Geospatial, temporal, and behavioural features are often created in the hospitality and travel industry. We created three new features for our model, which can be seen below.

Fig. 14. Creating Features (a)

```
#Creating two new columns in BnB_Df_unique called "avg_price_neigh" and "avg_price_city" which will contain
#the average price per neighbourbood and city.

avg_price_neigh = merged_BnB_Df.groupby('neighbourhood')['price'].transform('mean')
merged_BnB_Df['avg_price_neigh'] = merged_BnB_Df['price'] / avg_price_neigh
merged_BnB_Df['avg_price_neigh']

id
6.000000e+00    2.060265
1.090000e+02    0.768422
1.770000e+02    0.470457
3.600000e+02    0.781312
3.640000e+02    1.048134
                  ...
8.540000e+17    0.195568
8.540000e+17    0.196813
8.540000e+17    0.194024
8.540000e+17    0.195234
8.550000e+17    0.319604
Name: avg_price_neigh, Length: 9381527, dtype: float64

#Creating a new column in BnB_Df_unique called "avg_price_city" which will contain the average price per city.
avg_price_neigh = merged_BnB_Df.groupby('city')['price'].transform('mean')
merged_BnB_Df['avg_price_city'] = merged_BnB_Df['price'] / avg_price_neigh
merged_BnB_Df['avg_price_city']

id
6.000000e+00    2.048239
1.090000e+02    0.768948
1.770000e+02    0.466014
3.600000e+02    0.784555
3.640000e+02    1.055905
                  ...
8.540000e+17    0.194664
8.540000e+17    0.194191
8.540000e+17    0.194191
```

Fig. 15. Creating Features (b)

```
#Assuming that properties with more reviews have higher satisfaction rates as if the reviews
#were negative they would not get as many bookings and therefore not have a high number of reviews.
#create a feature called "customer_satisfaction"

merged_BnB_Df["customer_satisfaction"]=merged_BnB_Df["number_of_reviews"]/merged_BnB_Df["availability_365"]
```

### Encoding Categorical Variables:

Categorical variables can be encoded using techniques like one-hot encoding, label encoding, or target encoding, depending on the nature of the data and model requirements. By encoding categorical variables, you retain the information in these features while transforming them into a format suitable for mathematical computations. Unsupervised learning algorithms derive patterns, structures, or relationships within data. Encoding categorical variables enables these algorithms to understand and uncover hidden structures or groupings within the dataset. As we identified "room_type" as an essential categorical feature, we needed to encode it to improve usability. We chose One-Hot encoding over Labelling as there are no straightforward ways to weight the categories.

Fig. 16. One-Hot Encoding of "room_type"

```
In [5]:  #Using One-Hot Encoding on the nominal variable "room_type".
         merged_BnB_Df = pd.get_dummies(merged_BnB_Df, columns=['room_type'], prefix='property_type')
```

## Model Selection and Training

Our chosen unsupervised learning algorithm is K-means clustering. Clustering is an unsupervised learning method that divides data into natural groupings, and k-means clustering is one of the most basic but commonly used methods of unsupervised learning. K-means can group the data into any number of clusters using a distance measure. The benefit of clustering is that it is scalable, versatile, interpretable, and can be used on large datasets (Zubair et al., 2022).

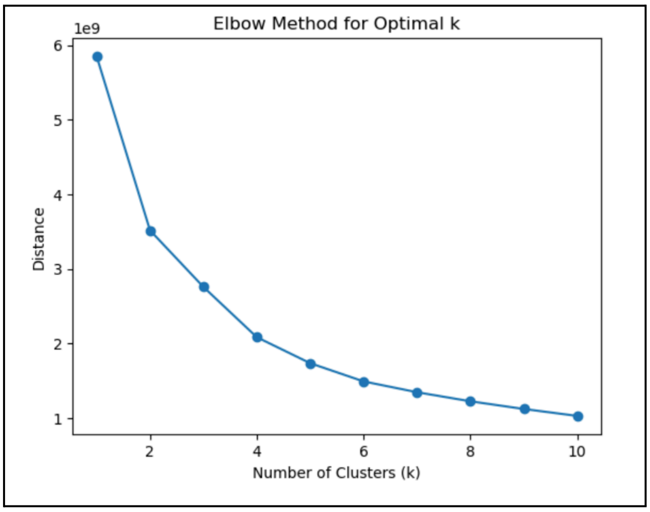
Clustering is highly beneficial for market segmentation and allows Airbnb to find similarities in property listings and discover what is essential to different customer groups in order to enhance consumers' experiences and increase profitability.

**Elbow Method:**

After cleaning and feature engineering, the first step is determining “k”, or the number of clusters the algorithm will create. The user must specify this number, so the Elbow method is used to find the optimal value. This method plots the Within-Sum of Squares (WWS value for each number of clusters). Within-Sum of Squares is the total absolute distance between the values and the cluster’s mean. The optimal number of k clusters is the value where k doesn’t remarkably change from the succeeding value and can be found at the point of inflection on the graph’s curve. If the number of clusters chosen is too large, it risks increasing the chances of overfitting the model (Malik & Tuckfield, 2019).

It is crucial to ensure that the k-means are run multiple times as the starting point for the cluster centres to reduce the risk of the centroid getting stuck and misconstruing the data. Within this example, we set the initialising runs at 10, meaning the code ran 10 times with different starting locations each time and selected the trail with the lowest WWS results. While this is important for the integrity of the results, it does take time and high computational power due to the size of our dataset. This issue will only increase with the size of the dataset.

Fig. 17. Using the Elbow Method for Optimal k



## Recommended Tools and Softwares

**Software Required and Libraries:**
The model employed in this report was implemented on an operating system that has Python 3 version 1.24.3 alongside Jupyter Notebook and Anacondas, as well as using the following libraries: Numpy, Pandas, Matplotlib, Seaborn, Sklearn and Anaconda as the main IDE.

**Recommended Softwares:**
Various unsupervised ML software programs are available within the travel and hospitality industry that could aid Airbnb hosts in recognising trends in property listings and help hosts establish reasonable rates. Two of the most prominent programs are Duetto and RateGain.

Duetto is primarily utilised in the hospitality sector as a method for revenue management. For Airbnb, revenue management aims to maximise profit by managing availability and price. Duetto would act as a powerful tool for Airbnb by using a clustering algorithm to identify poor performance periods and segment customer behaviours, therefore aiding in identifying trends previously unapparent. Duetto provides its revenue strategy programme through its "Command Centre" feature, in which hosts can immediately recognise and seize ample opportunities by clearly visualising their properties' revenue performance. Additionally, hosts can take immediate action by determining which days have unusually high or low demand and which need specific attention (McCay Tams, 2023). Host users can focus on properties or clusters that are over or under-performing by using customisable data views, enabling them to examine relevant data related to their listings (Duetto, 2016).

Another unsupervised ML software already successfully integrated by organisations within the travel and hospitality industry is RateGain. Similarly, RateGain offers pricing optimisation and revenue management solutions through the use of unsupervised ML in order to highlight competitor pricing approaches. This approach is visible in RateGain's MarketDRONE narrative feature, which aims to highlight key competitors' pricing models and optimise their pricing to maximise profit revenues. This feature provides the user with daily insights in order to precisely adjust their strategy and avoid losing market share to their rivals (RateGain, 2023)—moreover, the feature awards crucial time back to its users. No longer required to focus on finding hidden patterns within their data, users can primarily focus on attracting new consumers and increasing revenues.

**Identification of  ML Limitations**

Various limitations associated with machine learning can be identified by applying classical unsupervised learning with Airbnb's pricing strategy.

One of the most crucial limitations of any ML model is the quantity and quality of the data (Egger, 2022). Questions such as "how rich is our data?", "is it joinable?" and "has it been preprocessed correctly?" are key questions to ask at the data cleaning stage of building an ML model. More specifically, in the case of unsupervised learning, the lack of assessment abilities indeed calls for high-quality data (Egger, 2022). To amplify the importance of quality data, the phrase "garbage in - garbage out" is commonly used to exhaust the importance of obtaining rich and diverse data to predict outcomes or find hidden patterns within the data accurately.

Another key problem with ML is underfitting or overfitting the chosen model. Through underfitting, the model lacks complexity and will fail to analyse relationships within data or make accurate predictions, leading to unrecognised critical features and trends. Overfitting is the opposite, in which the model becomes too complex and fits the training set "too perfectly", resulting in a highly sensitive model that performs exceptionally poorly on unseen data. Additionally, it is important to mention that despite a model perhaps only giving a 5% improvement, companies must focus on what matters most and realise that perhaps that 5% in the big picture could mean substantial progress for the company (Bulanov, 2023).

Often termed the "black box" problem, another critical limitation in ML is the difficulty of interpretation (Egger, 2022). This issue is particularly relevant in the context of Airbnb's diverse property listings, where understanding how pricing recommendations are generated is crucial for trust and regulatory compliance. In addition, the interpretability of results goes hand in hand with knowing your product (Bulanov, 2023). It is essential in its implementation to understand how to use the algorithm to the company's advantage and align it with its mission, vision and values.

Another challenge faced by companies implementing ML is the importance of considering the company's maturity and analysing the high computational costs that ML can incur. Implementing ML needs to be a well-thought-out decision for a company, especially in the introductory stage, as solid experience and knowledge are critical in truly understanding what it can do and how it can benefit the company. Outsourcing this knowledge or experience can become extremely expensive and potentially impact the company's scalability and cost-effectiveness. In the case of Airbnb, we would not deem this as a challenge as a global digitised platform-based business; they likely have sufficient skilled talent and funds to implement such technologies.

## Conclusion

Overall, this report focuses on how classical unsupervised learning can benefit organisations within the travel industry. Within this, we decided to focus on Airbnb's US listings and hone in on its pivotal challenge of getting hosts to price their listings accurately. In our first stage, data cleaning, we handled our missing values, performed standardisation and removed duplicates and outliers along with other techniques. Following this, we created new features for our model and encoded categorical variables using one-hot encoding. We then chose K-means clustering as our chosen algorithm and determined the optimal number of k clusters using the elbow method. Furthermore, we recommended two software programs, Duetto and RateGain, from which both the travel industry and Airbnb could benefit. Lastly, we identified potential limitations and challenges associated with unsupervised ML that could hinder a model's performance.

Regarding our report, two recommendations we deem potentially beneficial for Airbnb to implement would be introducing host training programs and enhancing its web interface, making it more host-friendly. Through host training programs, Airbnb's hosts could gain better insight into the factors influencing listing prices and gain front-end knowledge from Airbnb's pricing experts. In addition, hosts can share their thoughts and local knowledge with Airbnb, consequently improving the model's performance. An enhanced web interface highlighting price insights, suggested adjustments and justifications for them would enable hosts to use the model better. Moreover, a feedback section could also be employed, allowing hosts to give their input on price accuracy, thus improving Airbnb's model and making it better tuned to the unique preferences of the host.

Regarding potential avenues for future research, Airbnb could integrate additional external sources into the unsupervised model to enhance the model's ability further. These indicators could include local events, weather forecasts or any PESTLE factors. Alongside this, Airbnb could investigate the pre-ML decision-making progress of its hosts when setting property prices. What hosts consider when setting prices and what sources they use could prove valuable sources of information when analysing how hosts make decisions in the absence of an unsupervised ML model.

## Data Availability

The dataset used in this report, compiled initially from multiple datasets found on Inside Airbnb, was downloaded from Kaggle. The most recent recompilation and updates were on the 14th of April, 2023. For our report, this dataset has undergone a lot of processing through Jupyter Notebook using Python 3.

## Files Employed

| | |
|---|---|
| Data Cleaning Code: | **US_BNB_2023_Cleaning.py (Google Drive)** |
| Feature Engineering Code: | **US_BNB_2023_Features.py (Google Drive)** |
| CSV File pre Data Cleaning: | **US_BNB_2023.csv (Google Drive)** |
| CSV File pre Feature Engineering: | **US_BNB_2023_Cleaned.csv (Google Drive)** |
| Model Training Code: | **US_BNB_2023_K-Means.py (Google Drive)** |

# References

Almasi, S. et al. (2023, September 27). The promise of travel in the age of AI. McKinsey. Retrieved 17 November 2023, from https://www.mckinsey.com/industries/travel-logistics-and-infrastructure/our-insights/the-promise-of-travel-in-the-age-of-ai#/

Bamisaye, T., & Alabi, O. (2023). Solving Key Business Challenges For A Client’s E-Commerce Using Ml Techniques. Retrieved 17 November 2023, from https://www.researchgate.net/profile/Oluwaseyi-Alabi-3/publication/372344526_Solving_Key_Business_Challenges_For_A_Client's_E-_Commerce_Using_Ml_Techniques/links/64b0f74595bbbe0c6e31ff9d/Solving-Key-Business-Challenges-For-A-Clients-E-Commerce-Using-Ml-Techniques.pdf

Bulanov, O. (2023, November 21). The Benefits of Using ML & AI in the Travel Industry. Django Stars. Retrieved 17 November 2023, from https://djangostars.com/blog/benefits-of-the-use-of-machine-learning-and-ai-in-the-travel-industry/

Chiny, M. et al. (2021a). A Client-Centric Evaluation System to Evaluate Guest’s Satisfaction on Airbnb Using Machine Learning and NLP. Applied Computational Intelligence and Soft Computing, 2021, e6675790. https://doi.org/10.1155/2021/6675790

Chiny, M. et al. (2021b) Towards a Machine Learning and Datamining approach to identify customer satisfaction factors on Airbnb. 2021 7th International Conference on Optimization and Applications (ICOA), pp. 1–5. doi: 10.1109/ICOA51614.2021.9442657.

Dhillon, J. et al. (2021). Analysis of Airbnb Prices using Machine Learning Techniques. 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), pp.0297–0303. doi: 10.1109/CCWC51732.2021.9376144.

Duetto. (2016). Hotel management (Duluth, Minn.). Vol. 231(Issue 11), pp. 143. Questex, LLC.

Egger, R. (2022). Machine Learning in Tourism: A Brief Overview. In R. Egger (Ed.), Applied Data Science in Tourism. Tourism on the Verge. Springer International Publishing, pp. 85–107. https://doi.org/10.1007/978-3-030-88389-8_6

Malik, A., & Tuckfield, B (2019). Chapter 1 Introduction to Clustering Methods. In Applied unsupervised learning with R: Uncover hidden relationships and patterns with K-means clustering, hierarchical clustering, and PCA. Packt Publishing.

McCay Tams, S. (2023, November 6). Duetto Launches Advance, Delivering Real-Time Rate Optimization and First-to-Market Data Integrations. Retrieved 17 November 2023, from https://www.duettocloud.com/press-releases/duetto-launches-advance-delivering-real-time-rate-optimization-and-first-to-market-data-integrations

Mukhamediev, R. I. et al. (2022). Review of Artificial Intelligence and Machine Learning Technologies: Classification, Restrictions, Opportunities and Challenges. *Mathematics*, *10*(15), Article 15. https://doi.org/10.3390/math10152552

RateGain (2023). Reinvent Your Hotel Pricing Strategy. Retrieved 17 November 2023, from https://rategain.com/hotels/rate-intelligence/

Swetaseal. (2023, January 15). Machine Learning use cases in Tourism. Medium. Retrieved 17 November 2023, from https://medium.com/@swetaseal/machine-learning-use-cases-in-tourism-f580d2f14c5b

Venture Beat. (2017, June 14). Airbnb VP talks about AI's profound impact on results. Retrieved 17 November 2023, from https://venturebeat.com/ai/airbnb-vp-talks-about-ais-profound-impact-on-profits/

Walker, M. (2022). Data cleaning and exploration with machine learning: Get to grips with machine learning techniques to achieve sparkling-clean data quickly. Packt Publishing.

WTTC. (2022, April 21). Travel & Tourism sector expected to create nearly 126 million new jobs within the next decade. World Travel & Tourism Council. Retrieved 17 November 2023, from https://wttc.org/news-article/travel-and-tourism-sector-expected-to-create-nearly-126-million-new-jobs-within-the-next-decade#:~:text=Julia%20Simpson%2C%20WTTC%20President%20%26%20CEO,be%20related%20to%20our%20sector.

Zhu, L. et al. (2019). Determinants of peer-to-peer rental rating scores: The case of Airbnb. International Journal of Contemporary Hospitality Management, 31(9), pp.3702–3721. https://doi.org/10.1108/IJCHM-10-2018-0841

Zubair, M. et al. (2022). An improved K-means Clustering Algorithm Towards an Efficient Data-Driven Modeling. Annals of Data Science. https://doi.org/10.1007/s40745-022-00428-2