CCT College Dublin

Module Title:	Problem Solving for the Industry
Assessment Title:	DEFINING FINANCIAL RISKS AND MARKET TRENDS THROUGH PREDICTIVE DATA ANALYSIS
Lecturer Name:	Dr Muhammad Iqbal
Students Full Name:	Giovanni Andrade Silva Luciana Teixeira Marcelle Louise de Souza Muhammad Shahbaz
Students Numbers:	2018320 2021322 2021381 2018092
Assessment Due Date:	16/05/2022
Date of Submission:	16/05/2022

Declaration

By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Table of Content

Acknowledgements

We wish to express our sincere gratitude to the professors in CCT, and particularly to our supervisor, Dr Muhammad Iqbal. His guidance and mentorship made this experience extremely memorable and rewarding to all of us.

The completion of this undertaking could not have been possible without the participation and assistance of the members of this group. Each one's contributions are sincerely appreciated and gratefully acknowledged.

To all our relatives, friends or others who in one way or another shared their support, either morally, financially, and physically, thank you very much.

List of Figures

List of Tables

Abstract

This project aims to use Machine Learning techniques and online property website datasets to determine financial risks and market trends for real estate investors in Ireland. By evaluating property-related data, this study seeks to identify patterns that may impact or aid these users in making smarter and cost-effective decisions. The concept of this research is to show the financial feasibility of data services, as well as how data science can improve business and operational efficiency.

1. Introduction

In recent years, natural catastrophes, terrorism attacks, socio-political conflicts, and, most recently, the COVID-19 outburst, all have had an impact on the tourism business. When Covid-19 restrictions prevented travel, it was no surprise that the short-term rental industry suffered (New York Times, 2021), a study conducted by LendingTree showed that listings dropped between March 2020 and March 2021 (Miller, 2022).

Despite the challenges posed by the pandemic and the government's first response to the problem (Irish Mirror, 2020), a study conducted by STR and AirDNA [1] found that house rentals outperformed hotels; among the most renowned platforms is Airbnb, since its founding in 2008 [2], Airbnb has become a symbol of the sharing economy and has changed the way people travel.

Airbnb hosts can establish their own prices, which poses a challenge because they must profit while retaining their popularity. Having data and relying on data analysis to spot trends and machine learning models to improve forecasts can be advantageous in this regard [3].

This project contains 255 features combined over six different datasets. Here we only list a few of them that are both representative and important for the task at hand: availability (minimum_night, maximum_night, availability_365, …); location (latitude, longitude, neighbourhood, area, …); financial (price; fee, extra guest fee, cleaning fee, daily fee…); reviews (reviews_per_month; comments, number of reviews, …).

This paper focuses on the Dublin short term rental market and our team has developed pricing and rate occupancy prediction models based on machine learning approaches (Linear Regression, Lasso, Ridge and K-Nearest Neighbour) as well as a Sentiment Analysis, in the field of NLP, of the reviews using Tensorflow libraries.

2. Research Methodology

CRISP-DM or Cross Industry Standard Process for Data Mining is a process model with six phases that naturally describes the data science life cycle. It's like a set of guardrails to help you plan, organise, and implement your data science or machine learning project [4].

A data science project must have a robust and predictable approach so people with limited data science experience can easily follow and comprehend its phases and results. [5] This is where the value of this framework becomes apparent, as you can apply this methodology as a template to guarantee you've examined all of the distinct issues unique to your project.

As visible on Figure 1, there are six phases involved in the process:

- Business Understanding
- Data Understanding
- Data Preparation
- Modelling
- Evaluation
- Deployment

Each of the six CRISP framework phases will be discussed in this report.

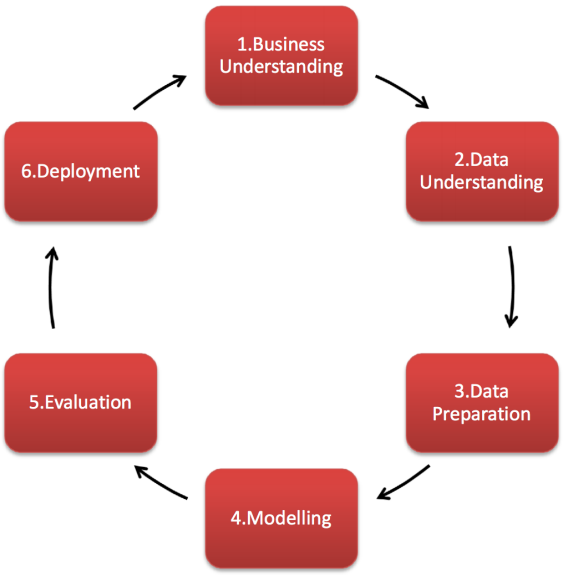


Figure 1: *CRISP Framework (Smart Vision - Europe, 2017)*

3. Business Understanding

The first phase of CRISP-DM focuses on bringing clarity to the business problem and thereby understanding what you want to accomplish from a business perspective. Neglecting this step can mean that a great deal of effort is put into producing the right answers to the wrong questions [6].

3.1 Objectives

The primary goal of this project is to provide people with a platform in the form of an application that allows them to search for properties based on their needs. People will no longer have to browse multiple websites and real estate agents to find out exactly the worth of a property; instead, they will have access to a single application that will provide them with current market prices as well as future estimates for a certain area. Initially, the application is intended to cover only properties in Dublin and its surroundings.

This data-driven application's future will not only be responsible for projecting pricing, but will also be able to propose destinations once users select filters in their searches. The filters can display properties that fit users budget, their needs, and general lifestyle.

3.2 Situation Assessment

Situation assessment helps to identify the challenges and opportunities internally and externally to your organisation, products and services. [7] In the current economic environment, there are various websites and applications where you can find properties for sale, rent, or book for short term accommodation, such as Daft.ie and Booking.com.

Unlike the other tools described, the purpose of this application is to collect data, analyse it, and make predictions, therefore, it cannot be labelled a competitor. This is due, in part, to the fact that this application will focus on displaying industry patterns, such as the connection between the occupancy rate and the yearly revenue. As a result, this information can help investors identify a suitable property, whereas information supplied on other websites, such as Daft.ie and Booking.com focuses fully on selling, renting or booking a property.

3.3 Data Mining Goals

The datasets Listings, Review and Rate were specifically chosen for predictions, while the datasets Calendar, Airbnb_listings1 and Airbnb_listing2 were dedicated to discover trends and patterns. Furthermore, we hope to meet the following conditions by utilising these datasets:

- Analyse preliminary data to determine the most appropriate modelling techniques for each scenario.
- Prepare data in accordance with standards.
- Ensure predictions achieve high accuracy from up to 80% or more.
- Evaluate trends and see how they affect predictions.

4. Data Understanding

In order to gain proper information from a dataset, the data set must be obtained and cleaned appropriately. Data understanding has two main phases: data assessment and data exploration. Data assessment is an important step in any project's development, as it can help identify the potential of the data and provide an estimate of its feasibility; simply meaning that during the data assessment phase, an organisation will gather information about the data and identify its potential.

This project uses Jupyter Notebook, an open-source web app that allows users to create interactive documents. It combines the input and output of various programs into a single file. Jupyter Notebook is an interactive web app that allows users to create documents that combine the input and output of various programs into a single file [8]. It is possible to determine data types, columns, non and null values, and memory use by utilising some of the methods accessible in this software, such as info(), describe(), head(), tail(), isnull().sum() and shape [9].

The six datasets present on this project are as follows:

1. Listing.csv
2. Reviews.csv
3. Calendar.csv
4. Airbnb-listings.csv
5. Airbnb-opendata.csv
6. Rate.csv

4.1 Dataset 1: Listing

The first dataset Listing, as represented on Figure 2 was sourced from the website insideairbnb.com [10] and here are some of the most relevant information:

- 6977 entries and 18 features;
- The data types present are: int64, float and object;
- The columns “neighbourhood_group” and “licence” presented only missing values, resulting in 11.11% of the dataset; these features were later removed;
- The whole dataset contained 16865 null values (“neighbourhood_group”, “licence”, “name”, “host_name”, “last_review” and “reviews_per_month”, approximately 13.42%;
- Memory usage of 981.3+ KB

```
In [9]: listing.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6977 entries, 0 to 6976
Data columns (total 18 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   id                              6977 non-null   int64
 1   name                            6976 non-null   object
 2   host_id                         6977 non-null   int64
 3   host_name                       6968 non-null   object
 4   neighbourhood_group             0 non-null      float64
 5   neighbourhood                   6977 non-null   object
 6   latitude                        6977 non-null   float64
 7   longitude                       6977 non-null   float64
 8   room_type                       6977 non-null   object
 9   price                           6977 non-null   int64
 10  minimum_nights                  6977 non-null   int64
 11  number_of_reviews               6977 non-null   int64
 12  last_review                     5527 non-null   object
 13  reviews_per_month               5527 non-null   float64
 14  calculated_host_listings_count  6977 non-null   int64
 15  availability_365                6977 non-null   int64
 16  number_of_reviews_ltm           6977 non-null   int64
 17  license                         0 non-null      float64
dtypes: float64(5), int64(8), object(5)
memory usage: 981.3+ KB
```

Figure 2. *Listing Dataset*

4.2 Dataset 2: Review

The second dataset Review, as represented on Figure 3 was also sourced from the website insideairbnb.com [10] and here are some of the most relevant information:

- 211213 entries and 6 features;
- The data types present are: int64 and object;
- The column “comments” presented 108 missing values, resulting in 0.008% of the dataset;
- This dataset is specifically for a Sentiment Analysis (see chapter 6.6) and the relevant columns are only comments and an added rating column;
- Subsets of this dataset needed to be created to support the analysis due to size;
- Memory usage of 9.7+ MB.

```
In [14]: review.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 211213 entries, 0 to 211212
Data columns (total 6 columns):
 #   Column         Non-Null Count   Dtype 
---  ------         --------------   ----- 
 0   listing_id     211213 non-null  int64 
 1   id             211213 non-null  int64 
 2   date           211213 non-null  object
 3   reviewer_id    211213 non-null  int64 
 4   reviewer_name  211213 non-null  object
 5   comments       211105 non-null  object
dtypes: int64(3), object(3)
memory usage: 9.7+ MB
```

Figure 3. *Review Dataset*

4.3 Dataset 3: Calendar

The third dataset Calendar, as represented on Figure 4 was also sourced from the website insideairbnb.com [10] and here are some of the most relevant information:

- 2546600 entries and 7 features;
- The data types present are: int64, float and object;

- The columns “price”, “adjusted_price”, “minimum_nights” and “maximum_nights” presented 382 null values, resulting in 0.002% of the dataset;
- Price and adjusted price feature have same values;
- The price feature was converted into a numeric value;
- Memory usage of 136.0+ MB.

```
calendar.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2546600 entries, 0 to 2546599
Data columns (total 7 columns):
 #   Column          Dtype  
---  ------          -----  
 0   listing_id      int64  
 1   date            object 
 2   available       object 
 3   price           object 
 4   adjusted_price  object 
 5   minimum_nights  float64
 6   maximum_nights  float64
dtypes: float64(2), int64(1), object(4)
memory usage: 136.0+ MB
```

Figure 4. *Calendar Dataset*

4.4 Dataset 4: Airbnb-listings

The fourth dataset Airbnb-listings, as represented on Figure 5. was sourced from the website public.opendatasoft.com [11] and here are some of the most relevant information:

- This dataset was named “airbnb_listing_1” and later used in combination with the fifth dataset;
- 8151 entries and 89 features;
- The data types present are: int64, float and object;

- The columns “Name”, “Calculated host listings count”, “Reviews per Month” and “Features” presented 1669 null values, resulting in 0.2% of the dataset;
- The price feature was converted into a numeric value;
- Memory usage of 5.5+ MB.

```
In [23]: airbnb_listing_1.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8151 entries, 0 to 8150
Data columns (total 89 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   ID                              8151 non-null   int64
 1   Listing Url                     8151 non-null   object
 2   Scrape ID                       8151 non-null   int64
 3   Last Scraped                    8151 non-null   object
 4   Name                            8149 non-null   object
 5   Summary                         8019 non-null   object
 6   Space                           5396 non-null   object
 7   Description                     8150 non-null   object
 8   Experiences Offered             8151 non-null   object
 9   Neighborhood Overview           4794 non-null   object
 10  Notes                           3389 non-null   object
 11  Transit                         5015 non-null   object
 12  Access                          4929 non-null   object
 13  Interaction                     4699 non-null   object
 14  House Rules                     5149 non-null   object
 15  Thumbnail Url                   7407 non-null   object
 16  Medium Url                      7407 non-null   object
 17  Picture Url                     8137 non-null   object
 18  XL Picture Url                  7407 non-null   object
 19  Host ID                         8151 non-null   int64
```

Figure 5. *Airbnb-listings Dataset*

4.5 Dataset 5: Airbnb-opendata

The fifth dataset Airbnb-opendata, as represented on Figure 6 was also sourced from the website public.opendatasoft.com [11] and here are some of the most relevant information:

- This dataset was named “airbnb_listing_2” and later used in combination with the fourth dataset;
- 4263 entries and 89 variables;
- The data types present are: int64, float and object;
- The columns “Name”, “Reviews per Month” and “Features” presented 854 null values, resulting in 0.2% of the dataset;

- Memory usage of 2.9+ MB.

```
In [24]: airbnb_listing_2.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4263 entries, 0 to 4262
Data columns (total 89 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   ID                              4263 non-null   int64
 1   Listing Url                     4263 non-null   object
 2   Scrape ID                       4263 non-null   int64
 3   Last Scraped                    4263 non-null   object
 4   Name                            4262 non-null   object
 5   Summary                         4219 non-null   object
 6   Space                           2685 non-null   object
 7   Description                     4262 non-null   object
 8   Experiences Offered             4263 non-null   object
 9   Neighborhood Overview           2343 non-null   object
 10  Notes                           1664 non-null   object
 11  Transit                         2440 non-null   object
 12  Access                          2469 non-null   object
 13  Interaction                     2326 non-null   object
 14  House Rules                     2623 non-null   object
 15  Thumbnail Url                   3851 non-null   object
 16  Medium Url                      3851 non-null   object
 17  Picture Url                     4254 non-null   object
 18  XL Picture Url                  3851 non-null   object
 19  Host ID                         4263 non-null   int64
 20  Host URL                        4263 non-null   object
```

Figure 6. *Airbnb-opendata Dataset*

4.6 Dataset 6: Rate

The sixth dataset Rate, as represented on Figure 7. was also sourced from the website app.airbtics.com [12] and here are some of the most relevant information:

- The dataset was a sample, therefore it was presented unstructured meaning columns needed to rearranged and named so that an accurate overview could be provided;
- 3024 entries and 46 variables;
- The data type present is only object;
- There was no null values on the dataset;
- Numeric columns were later encoded to float or int64;
- Memory usage of 1.1+ MB.

```
In [311]: rate.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 3024 entries, 4 to 3027
Data columns (total 46 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Listing URL                3024 non-null   object
 1   Property Type              3024 non-null   object
 2   Latitude                   3024 non-null   float64
 3   Longitude                  3024 non-null   float64
 4   Star Rating                3024 non-null   float64
 5   Number of Active Days      3024 non-null   int64
 6   Bedrooms                   2866 non-null   float64
 7   Has pool                   3024 non-null   object
 8   Cleaning Fee               3024 non-null   int64
 9   Extra Guest Fee            3024 non-null   float64
 10  Daily Rate (2020-11)       3024 non-null   int64
 11  Daily Rate (2020-12)       3024 non-null   int64
 12  Daily Rate (2021-01)       3024 non-null   int64
 13  Daily Rate (2021-02)       3024 non-null   int64
 14  Daily Rate (2021-03)       3024 non-null   int64
 15  Daily Rate (2021-04)       3024 non-null   int64
 16  Daily Rate (2021-05)       3024 non-null   int64
 17  Daily Rate (2021-06)       3024 non-null   int64
 18  Daily Rate (2021-07)       3024 non-null   int64
 19  Daily Rate (2021-08)       3024 non-null   int64
 20  Daily Rate (2021-09)       3024 non-null   int64
```

Figure 7. *Rate Dataset*

5. Data Preparation

Most machine learning algorithms require data to be formatted in a certain way, therefore datasets must be prepared before they can produce useful insights. To reach the last step of preparation, the data must be cleaned, formatted, and changed to handle missing or incorrect values. If this is not the case, a prediction method might be unsuccessful. Data preparation includes combining and/or separating variables and columns, modifying data formats, removing unnecessary or duplicated data, and making changes to the dataset.

Data preparation was carried out for the six datasets in order to assure data quality and to picture the sort of modelling that would be performed for prediction.

5.1 Dataset 1: Listing

5.1.1 Data Cleaning

- Drop Columns

After analysing this dataset, it became evident that several of the columns were irrelevant to the prediction model phase. Therefore, the first step taken with this dataset was to drop these columns. As seen on Figure 8 below, five columns were dropped: id, host_name, last_review, neighbourhood_group and licence. These columns presented numerical and categorical values, however, they won't be needed for future predictions.

```
# Drop the data that are not of interest and/or causing privacy issues
listing.drop(['id','host_name','last_review'], axis=1, inplace=True)
```

```
listing.drop(['neighbourhood_group', 'license'], axis=1, inplace=True)
```

Figure 8. *Dataset Listings - Dropping columns*

- Outliers

A box plot was produced within the price limit to see if this dataset had any outliers (2000). This representation, as shown in Figure 9, verifies the occurrence of outliers. The approach known as IQR (Interquartile Range) was used to deal with these values.

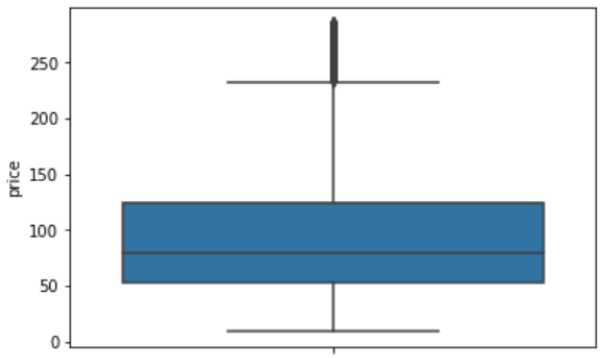


Figure 9. *Dataset Listings - Dealing with Outliers I*

The interquartile range shows how the data is spread about the median. It is less susceptible than the range to outliers and can, therefore, be more helpful. [13] After removing the outliers, the price distribution within those neighbourhoods becomes more realistic, as shown in Figure 10.

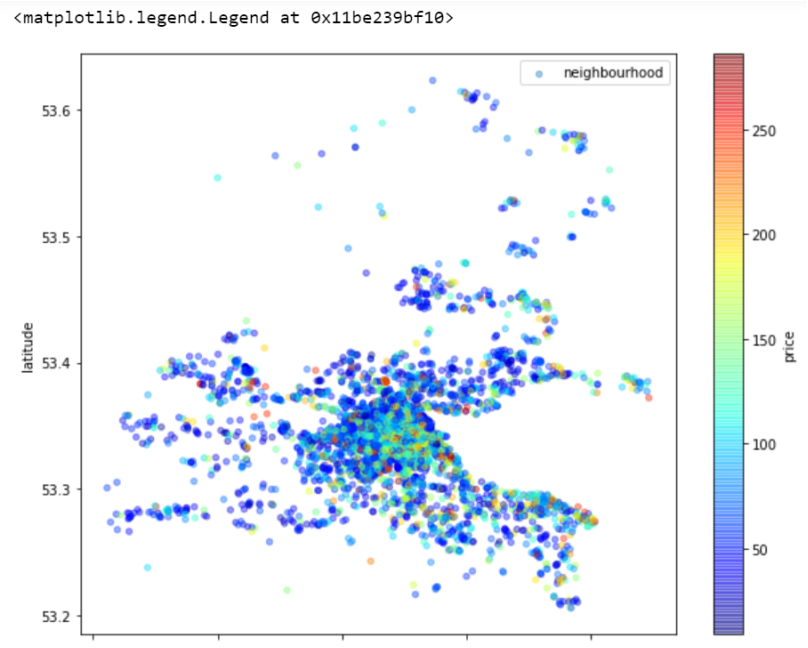


Figure 10. *Dataset Listings - Dealing with Outliers II*

- Null Values

The only columns with null values in this dataset are name (1) and reviews per month (1450). To appropriately deal with these values, we must first remember that reviews_per_month is a float variable. Sometimes rather than dropping NA values, you'd rather replace them with a valid value. [14]

Taking that into consideration, the fillna() function can be used to replace these null values with a given value, which in this case was 0. As seen in Figure 11, after calling this function, null values are no longer recognised in the column reviews_per_month.

```
listing.fillna({'reviews_per_month':0}, inplace=True)
```

```
listing.isnull().sum()
```

```
name                              1
host_id                           0
neighbourhood                     0
latitude                          0
longitude                         0
room_type                         0
price                             0
minimum_nights                    0
number_of_reviews                 0
reviews_per_month                 0
calculated_host_listings_count    0
availability_365                  0
number_of_reviews_ltm             0
dtype: int64
```

Figure 11. *Dataset Listings - Null Values After Cleaning*

- Data Encoding

To encode the dataset columns, pd.cut method was called: this function is also useful for going from a continuous variable to a categorical variable. [15] In Figure 12, some of these columns are being encoded to categorical.

```
# Recode data as categorical
listing_encoded = listing.copy()
listing_encoded['minimum_nights'] = pd.qcut(listing['minimum_nights'], q=2, labels=["minimum_nights_low", "minimum_nights_h
listing_encoded['number_of_reviews'] = pd.qcut(listing['number_of_reviews'], q=3, labels=["number_of_reviews_low", "minimum
listing_encoded['reviews_per_month'] = pd.qcut(listing['reviews_per_month'], q=2, labels=["reviews_per_month_low", "reviews
listing_encoded['calculated_host_listings_count'] = pd.cut(listing['calculated_host_listings_count'],
                                                bins=[0, 2, 327],
                                                labels=["calculated_host_listings_count_low", "calculated_host_listings_cou
```

Figure 12. *Dataset Listings - Encoding As Categorical Data*

5.1.2 Patterns and Trends

In this dataset, four specific patterns were discovered to have a higher probability of influencing the final predictions.

- Most popular neighbourhood

In Figure 13, Dublin City has the highest number of properties (5334) listed followed by Dun Laoghaire (742), Fingal (616), and South Dublin (285).

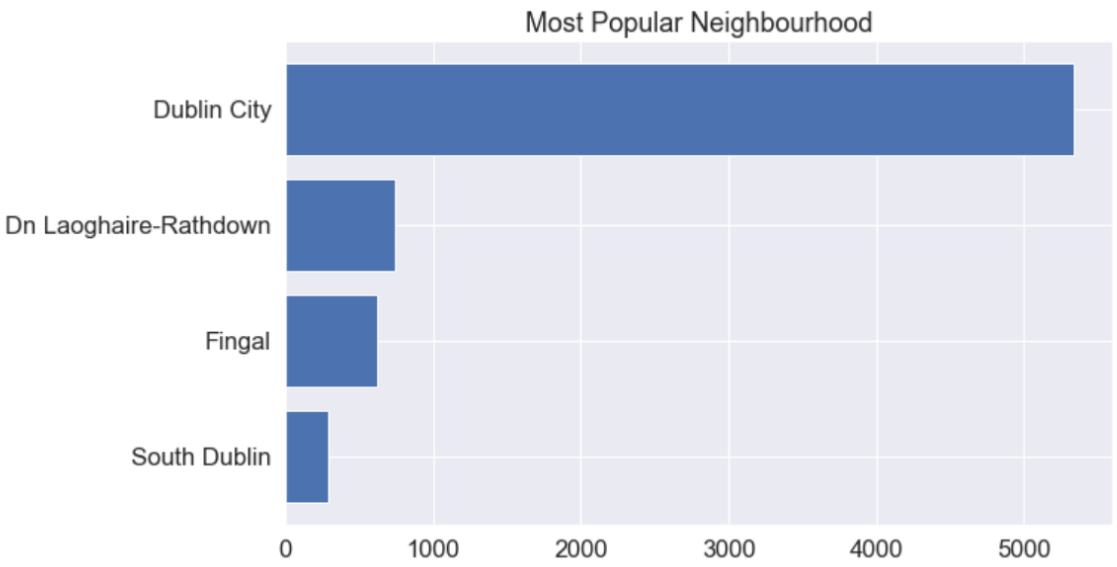


Figure 13. *Dataset Listings - Most Popular Neighbourhood*

- Most popular room type

As visible on Figure 14, entire homes or apartments appear to be the most popular type of property, followed by hotel rooms, private rooms, and shared rooms (hostels).

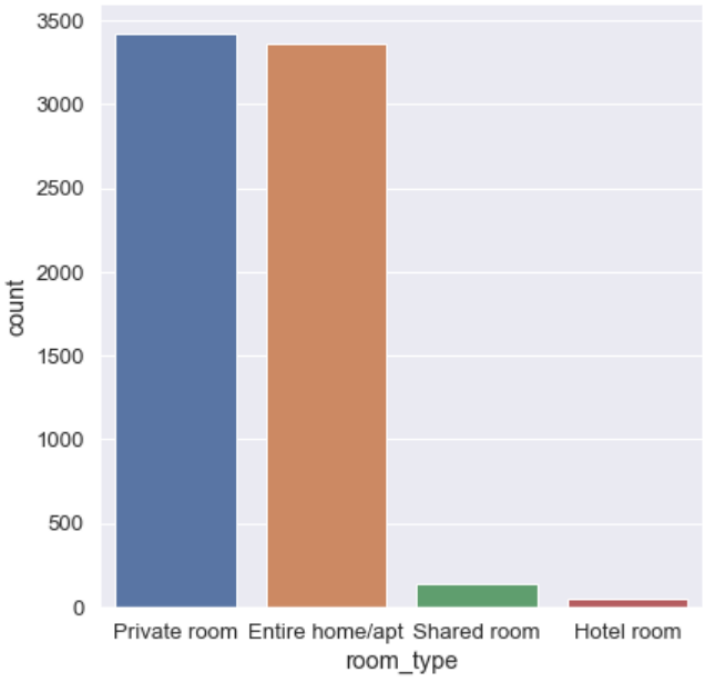


Figure 14. *Dataset Listings - Most Popular Room Types*

- Availability through the year

Out of 6977 only 2740 properties had more than one day available for rental, while 4237 did not have a single day available at the time this data was gathered. In addition, 186 properties were available at least 358 days every year.

Figure 15 shows that the properties available in Fingal have attained around/or more than 90% availability, followed by South Dublin (about 85%) and Dun Laoghaire (around 75%). However, properties in Dublin City are less likely to be available: with a rate of 60%, it's evident that this area is the most desired by guests.

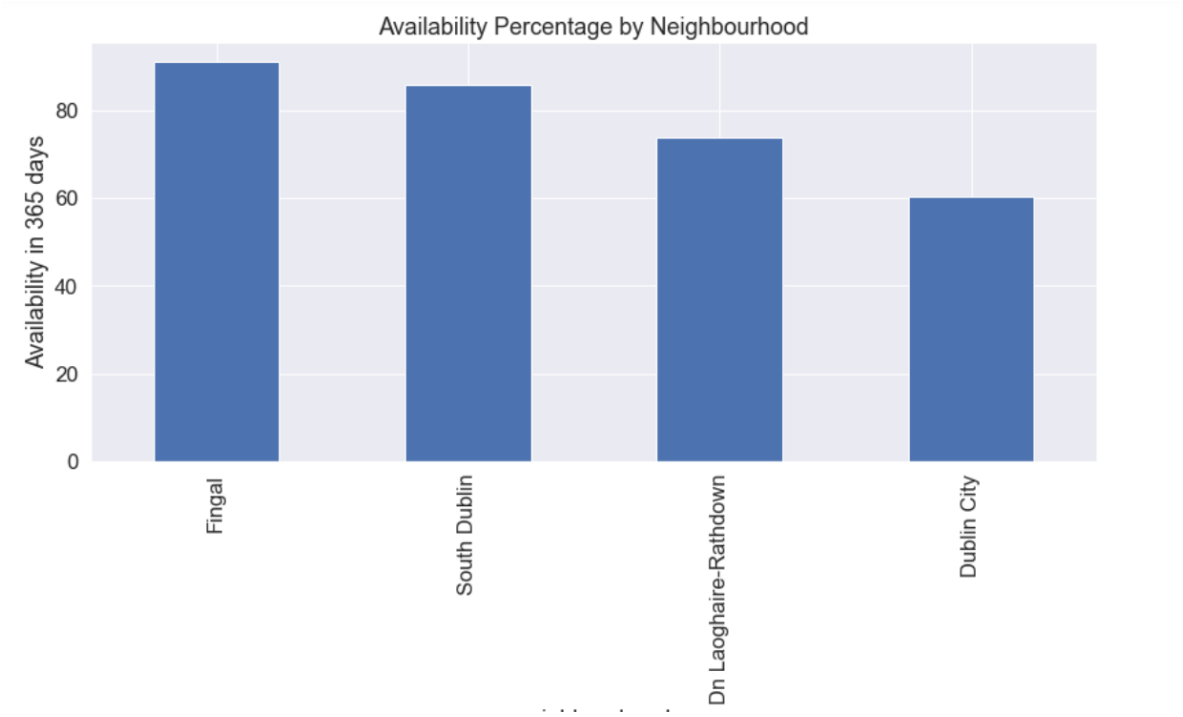


Figure 15. *Dataset Listings - Availability by Neighbourhood*

- Average price per neighbourhood

Considering the number of properties and low availability for Dublin City, there was an indication that this area would be the most expensive to rent in. To verify this hypothesis, the mean() was used to group the listings by neighbourhood and find out the average price for all neighbourhoods.

```
listing.groupby('neighbourhood')['price'].mean().sort_values(ascending=False)
```

```
neighbourhood
Dublin City              359.312148
Dn Laoghaire-Rathdown    121.846361
Fingal                   121.387987
South Dublin              88.729825
Name: price, dtype: float64
```

Figure 16. *Dataset Listings - Average Price by Neighbourhood I*

As previously stated, Dublin City has indeed the higher average price per night, as indicated in figure 16.

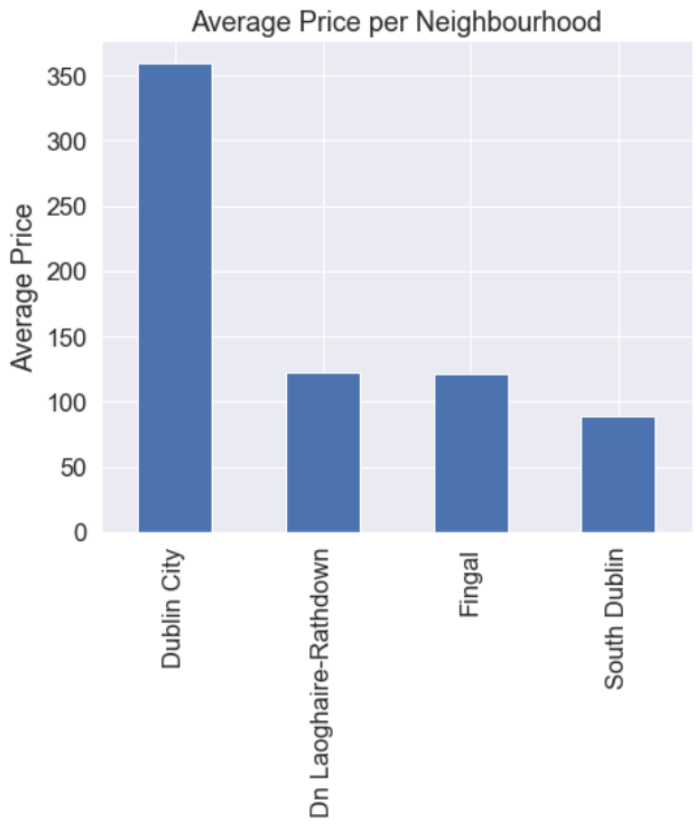


Figure 17. *Dataset Listings - Average Price by Neighbourhood II*

5.2 Dataset 2: Review

5.2.1 Data Cleaning

- Null Values

Figure 18 shows that just one of the six columns in this dataset has null values, which is the "Comments" column. Because this field does not represent numerical values, the quote "Not available" was used instead of replacing the null value with a 0.

```
review.isnull().sum()
```

```
listing_id         0
id                 0
date               0
reviewer_id        0
reviewer_name      0
comments         108
dtype: int64
```

```
review.fillna({'comments': 'Not available'}, inplace = True)
```

```
review.isnull().sum()
```

```
listing_id       0
id               0
date             0
reviewer_id      0
reviewer_name    0
comments         0
dtype: int64
```

Figure 18. *Dataset Review - Dealing with Null Values*

- Drop Columns

Since the prediction model will only impact the column "comments," the other columns are no longer required. Figure 19 shows that these columns were removed to make future modelling easier.

```
#Dropping columns that are not needed for the future model
review.drop(['id', 'listing_id','date','reviewer_id','reviewer_name'], axis=1, inplace=True)
```

```
review.head()
```

	comments
0	We enjoyed our stay very much. The room was co...
1	We have been here 4 nights. Stay in a home is ...
2	Teresa and Hughie were great hosts. They were ...
3	No surprises, was as described. Very gracious...
4	Teresa was a lovely hostess, and we had a deli...

Figure 19. *Dataset Review - Dropping Columns*

5.2.2 Patterns and Trends

Word Cloud is a data visualisation technique used for representing text data in which the size of each word indicates its frequency or importance [16]. Sentiment Analysis was used to identify the comments in this dataset and show the top words from the comments, as seen in Figure 20.

Figure 20. *Dataset Review - Top Words in Comments*

5.3 Dataset 3: Calendar

5.3.1 Data Cleaning

- Data Types

To perform any time series based operation on the dates if they are not in the right format. In order to be able to work with it, it is required to convert the dates into the datetime format [17]. The specified column is converted to datetime by executing the method pd.to datetime, as shown in Figure 21.

```
calendar['date'] = pd.to_datetime(calendar.date)
calendar.info(verbose=True, null_counts=True)
```

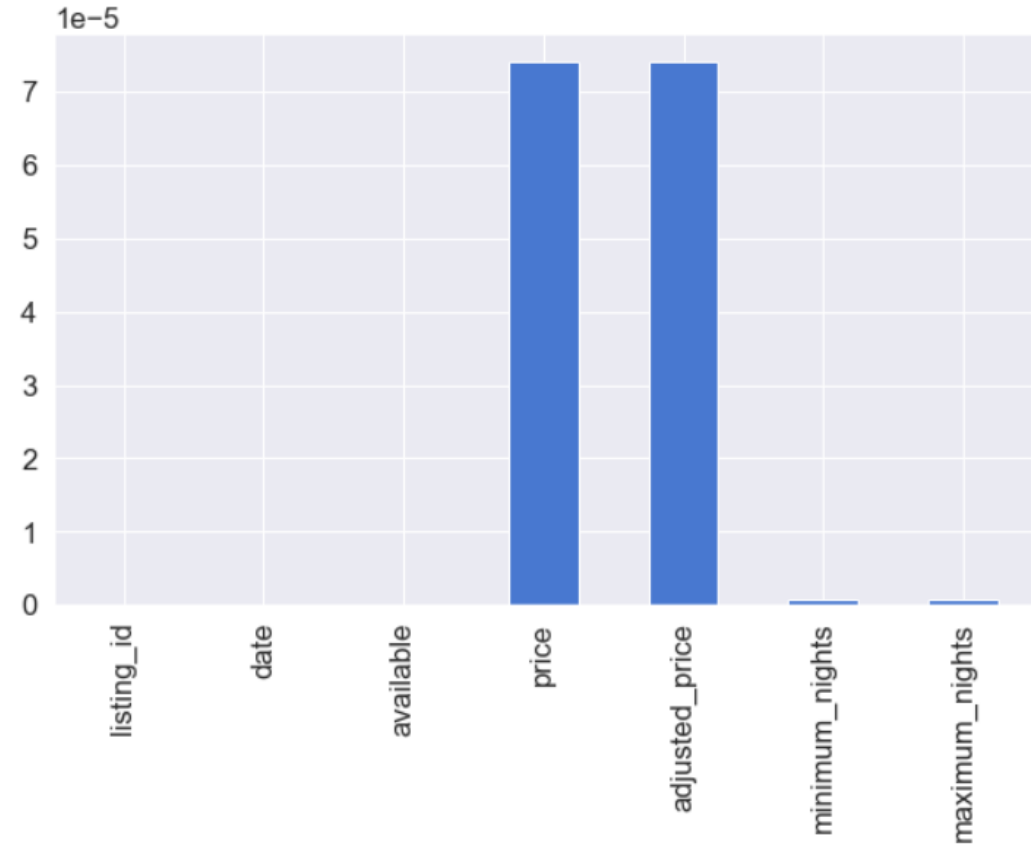
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2546600 entries, 0 to 2546599
Data columns (total 7 columns):
 #   Column          Non-Null Count    Dtype
---  ------          --------------    -----
 0   listing_id      2546600 non-null  int64
 1   date            2546600 non-null  datetime64[ns]
 2   available       2546600 non-null  object
 3   price           2546411 non-null  object
 4   adjusted_price  2546411 non-null  object
 5   minimum_nights  2546598 non-null  float64
 6   maximum_nights  2546598 non-null  float64
dtypes: datetime64[ns](1), float64(2), int64(1), object(3)
memory usage: 136.0+ MB
```

Figure 21. *Dataset Calendar - Data Types*

- Missing Values

As illustrated on Figure 22, the columns “price”, “adjusted_price”, “minimum_nights”, and “maximum_nights” contain missing data. At this time, only the column “price” was chosen to be treated. The.dropna() function was used to remove the null values of this column.

```
plt.figure(figsize=(10,6));
null_price = calendar.isnull().sum()
(null_price/calendar.shape[0]).plot(kind='bar');
```




```
calendar.dropna(axis=0,subset=['price'],inplace=True)
```

Figure 22. *Dataset Calendar - Missing Values*

- New Columns

The Month and Year columns were created to split the data that was previously accessible in the column "date" as seen on Figure 23. Breaking this data into different columns may result in more accurate predictions.

```
calendar['month'], calendar['year'] = calendar.date.dt.month, calendar.date.dt.year
calendar.info(verbose=True, null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2546411 entries, 0 to 2546599
Data columns (total 9 columns):
 #   Column          Non-Null Count    Dtype
---  ------          --------------    -----
 0   listing_id      2546411 non-null  int64
 1   date            2546411 non-null  datetime64[ns]
 2   available       2546411 non-null  object
 3   price           2546411 non-null  float64
 4   adjusted_price  2546411 non-null  float64
 5   minimum_nights  2546409 non-null  float64
 6   maximum_nights  2546409 non-null  float64
 7   month           2546411 non-null  int64
 8   year            2546411 non-null  int64
dtypes: datetime64[ns](1), float64(4), int64(3), object(1)
memory usage: 194.3+ MB
```

***Figure 23**. Dataset Calendar - Added Columns*

5.3.2 Patterns and Trends

- Availability vs average prices

In Figure 24, we can see that the majority of the properties are available for rental between 0 and 200 days. The blue dots indicate that the property is not available, while the orange dots indicate that it is.

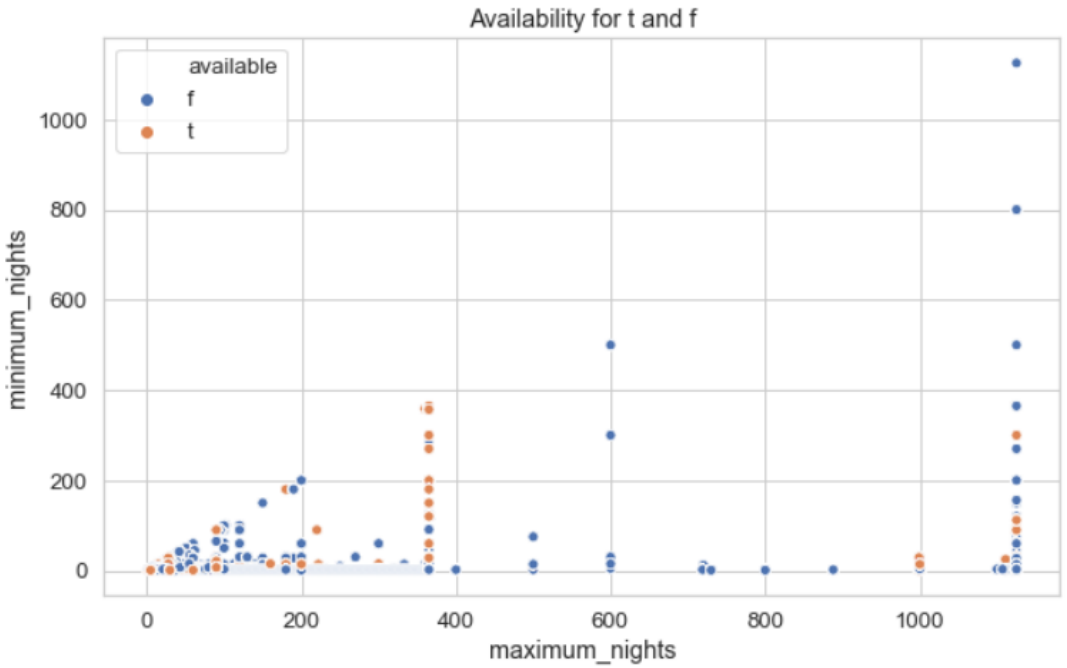


Figure 24. *Dataset Calendar - Patterns and Trends*

5.4 Dataset 4 and 5: Airbnb-Listings and Airbnb-OpenData

5.4.1 Data Cleaning

- Append

As indicated on chapter 4.5 of Data Understanding, these two datasets contain a large number of entries and are connected since they come from the same source. As a consequence, merging these could assist in achieving better prediction outcomes.

The append() method in Python adds a single item to a list. It does not return a new list of things, but rather adds the item to the end of the existing one. By looking at the dataset entries again on Figure 25 we can verify that the datasets have been successfully merged.

```
airbnb1 = pd.DataFrame(airbnb_listing_1)
airbnb2 = pd.DataFrame(airbnb_listing_2)
```

```
airbnb_final = airbnb1.append(airbnb2, ignore_index = True)
```

```
#Confirming that the rows have been added
airbnb_final.shape
```

```
]: (12414, 89)
```

Figure 25. *Dataset Airbnb Merged Datasets - Merging Datasets*

- Drop Columns

After the datasets were merged, some of these columns had to be eliminated due to the high number of duplicated values. Figure 26 shows the dropped columns.

```
airbnb_clean = airbnb_final.drop(columns=['Listing Url', 'Scrape ID', 'Last Scraped', 'Name', 'Summary',
       'Space', 'Description', 'Neighborhood Overview',
       'Notes', 'Transit', 'Access', 'Interaction', 'House Rules',
       'Thumbnail Url', 'Medium Url', 'Picture Url', 'XL Picture Url',
       'Host URL', 'Host Name', 'Calendar Updated', 'Calendar last Scraped',
       'Host About','Host Thumbnail Url', 'Host Picture Url', 'Host Verifications', 'Street',
       'Has Availability', 'Availability 30', 'Availability 60',
       'Availability 90', 'First Review', 'Last Review','License', 'Jurisdiction Names',
       'Cancellation Policy', 'Calculated host listings count'])
```

Figure 26. *Dataset Airbnb Merged Datasets - Dropping Columns*

- Drop Rows

Both datasets featured many locations in Ireland and the United Kingdom. Because this study was limited to Ireland, the data's accuracy was improved by removing any rows that reflected locations other than Dublin City, as seen on Figure 27.

```
airbnb_clean[airbnb_clean.NeighbourhoodCleansed != 'Dublin City']
```

Figure 27. *Dataset Airbnb Merged Datasets - Rows Dropped*

- Null Values

All null values in categorical data columns were replaced with 0 in this dataset, as seen on Figure 28.

```
airbnb_clean.fillna({'Host Location':0}, inplace=True)
airbnb_clean.fillna({'Host Response Time':0}, inplace=True)
airbnb_clean.fillna({'Host Acceptance Rate':0}, inplace=True)
airbnb_clean.fillna({'Host Neighbourhood':0}, inplace=True)
airbnb_clean.fillna({'Neighbourhood':0}, inplace=True)
airbnb_clean.fillna({'Neighbourhood Group Cleansed':0}, inplace=True)
airbnb_clean.fillna({'City':0}, inplace=True)
airbnb_clean.fillna({'State':0}, inplace=True)
airbnb_clean.fillna({'Zipcode':0}, inplace=True)
airbnb_clean.fillna({'Market':0}, inplace=True)
airbnb_clean.fillna({'Amenities':0}, inplace=True)
airbnb_clean.fillna({'Features':0}, inplace=True)
```

Figure 28. *Dataset Airbnb Merged Datasets - Null Values I*

However, replacing the present values of some of these columns with 0 would not be an appropriate option since this replacement might weaken the results of regression models. As a result, median was utilised to fill in the blanks for these specific columns, as seen on Figure 29, Median is considered to be a better technique in such occasions as it does not influence outliers.

Median has a very big advantage over Mean, which is the median value is not skewed so much by extremely large or small values. The median value is either contained in the data-set of values provided or it doesn't sway too much from the data provided [18].

```
missing_col = ['Host Response Rate']
for i in missing_col:
 airbnb_clean.loc[airbnb_clean.loc[:,i].isnull(),i]=airbnb_clean.loc[:,i].median()

missing_col = ['Bathrooms']
for i in missing_col:
 airbnb_clean.loc[airbnb_clean.loc[:,i].isnull(),i]=airbnb_clean.loc[:,i].median()

missing_col = ['Bedrooms']
for i in missing_col:
 airbnb_clean.loc[airbnb_clean.loc[:,i].isnull(),i]=airbnb_clean.loc[:,i].median()

missing_col = ['Beds']
for i in missing_col:
 airbnb_clean.loc[airbnb_clean.loc[:,i].isnull(),i]=airbnb_clean.loc[:,i].median()

missing_col = ['Square Feet']
for i in missing_col:
 airbnb_clean.loc[airbnb_clean.loc[:,i].isnull(),i]=airbnb_clean.loc[:,i].median()

missing_col = ['Price']
for i in missing_col:
 airbnb_clean.loc[airbnb_clean.loc[:,i].isnull(),i]=airbnb_clean.loc[:,i].median()

missing_col = ['Weekly Price']
for i in missing_col:
 airbnb_clean.loc[airbnb_clean.loc[:,i].isnull(),i]=airbnb_clean.loc[:,i].median()

missing_col = ['Monthly Price']
for i in missing_col:
 airbnb_clean.loc[airbnb_clean.loc[:,i].isnull(),i]=airbnb_clean.loc[:,i].median()
```

Figure 29. *Dataset Airbnb Merged Datasets - Null Values II*

- New Column

Because of the significant number of neighbourhoods in the dataset, a new column was necessary to group these values for future visualisations. Therefore, the column "area" was added, and these neighbourhoods were divided into their respective areas, as seen on Figure 30.

```
def getArea(Neighbourhood):
    if (Neighbourhood == "Artane" or Neighbourhood == "Fairview" or Neighbourhood == "Stoneybatter/Arbour Hill" or Neighbourh
       return "North Central Area"
    elif(Neighbourhood == "" or Neighbourhood == "Ballymun" or Neighbourhood == "Santry" or Neighbourhood == "Finglas"):
       return "North West Area"
    elif(Neighbourhood == "Clontarf" or Neighbourhood == "Drumcondra" or Neighbourhood == "Phibsborough"):
       return "North East Area"
    elif(Neighbourhood =="Ringsend/Irishtown" or Neighbourhood=="North City Central/O'Connell Street" or Neighbourhood == "Ol
       return "Central Area"
    elif(Neighbourhood =="Portobello" or Neighbourhood=="Ranelagh and Rathmines" or Neighbourhood == "Ballsbridge" or Neighbo
       return "South Central Area"
    elif(Neighbourhood =="Rathgar" or Neighbourhood=="Sandymount" or Neighbourhood == "Booterstown"):
       return "South East Area"
    elif(Neighbourhood =="Crumlin" or Neighbourhood=="Kimmage" or Neighbourhood == "Harold's Cross" or Neighbourhood == "Ball
       return "South West Area"
    else:
       return "undefined"


airbnb_dub['area'] = airbnb_dub['Neighbourhood'].apply(getArea)
airbnb_dub.head()
```

Figure 30. *Dataset Airbnb Merged Datasets - New Added Column*

5.4.2 Patterns and Trends

- Price vs Location

The plot, as shown in Figure 31, configures the relationship between the area or locations and the daily rate. With over 130 euros per night, the South East Area appears to have the highest rate, followed by the South Central Area (over 120 euros), and the Central Area (over 110 euros).

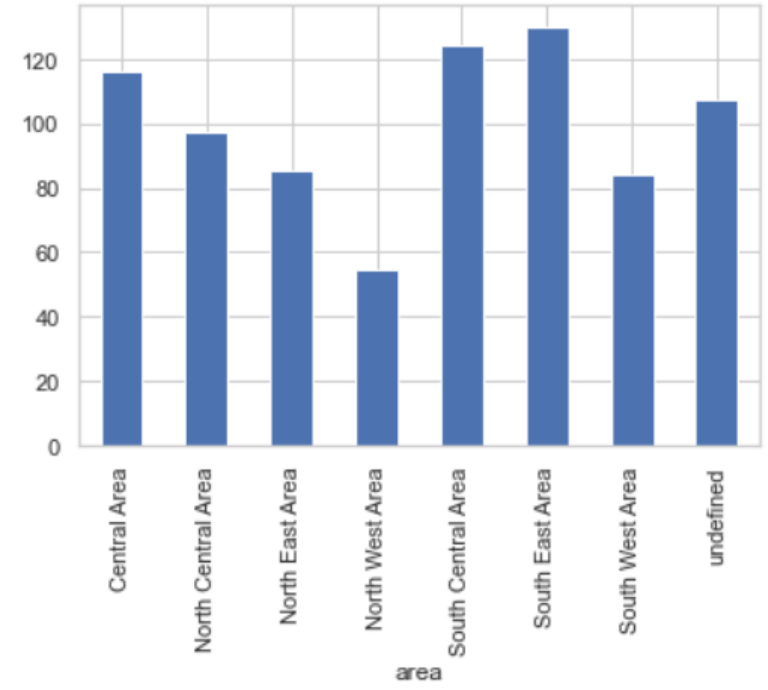


Figure 31. *Dataset Airbnb Merged Datasets - Patterns and Trends I*

- Price vs Room Type

Figure 32 illustrates that entire homes or apartments have higher nightly costs than private rooms and shared rooms.

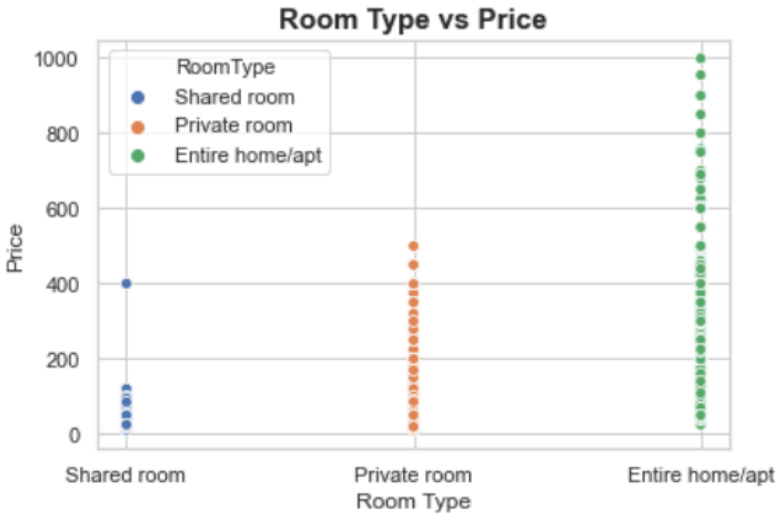


Figure 32. *Dataset Airbnb Merged Datasets - Patterns and Trends II*

- Most reviewed properties

The previous visualisations indicated that South Dublin was the area with the highest number of properties and price per night due to the high demand of guests. In Figure 33, the visualisation indicates that South Dublin also has the higher number of reviewed properties. The amount of properties with reviews is indicated by three colours on the map: orange, yellow, and green. Orange indicates the homes with the most reviews, yellow indicates the properties with the median number of reviews, and green indicates the properties with the fewest reviews.

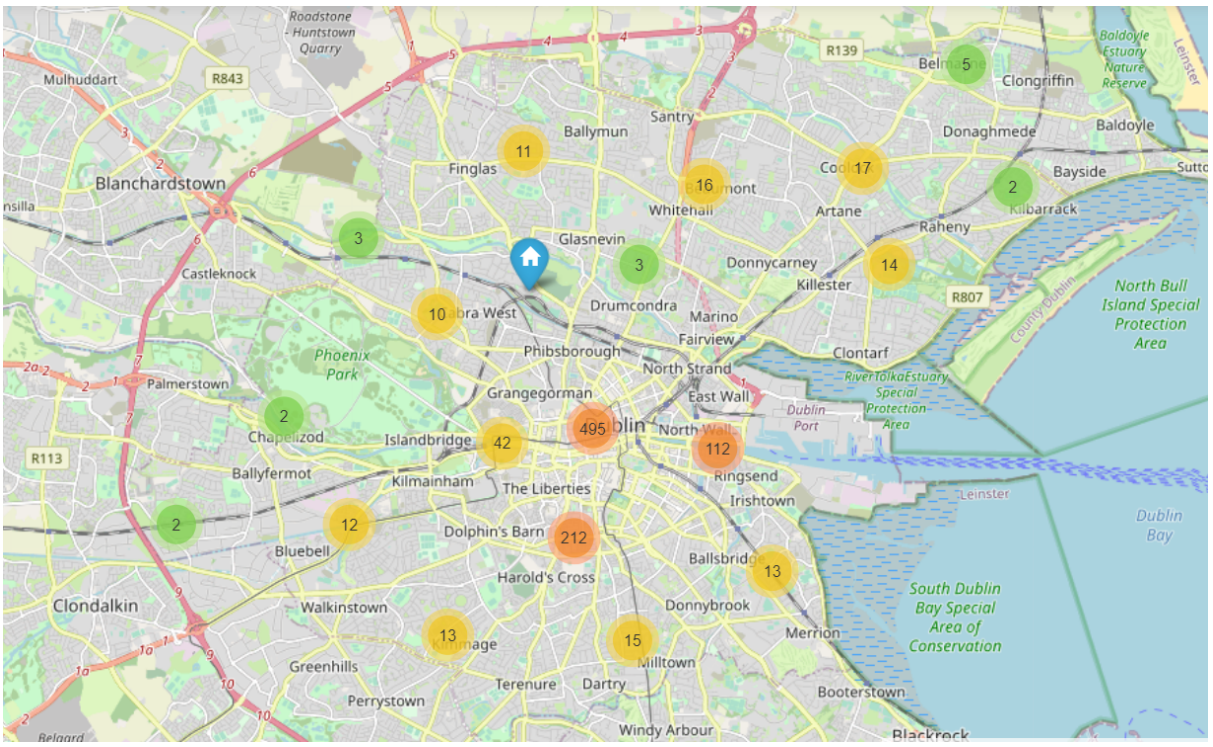


Figure 33. *Dataset Airbnb Merged Datasets - Patterns and Trends III*

- Area vs Room Type

According to the statistics shown in Figure 34, the North East area has the greatest number of homes with shared rooms. In respect to private rooms, the Central Area had the greatest instances, followed by the North Central Area and the North East Area. In the Central Area, entire homes or apartments have exhibited the highest number of properties, followed by the North Central Area and the South Central Area. This visualisation also highlights the large number of properties with unknown areas.

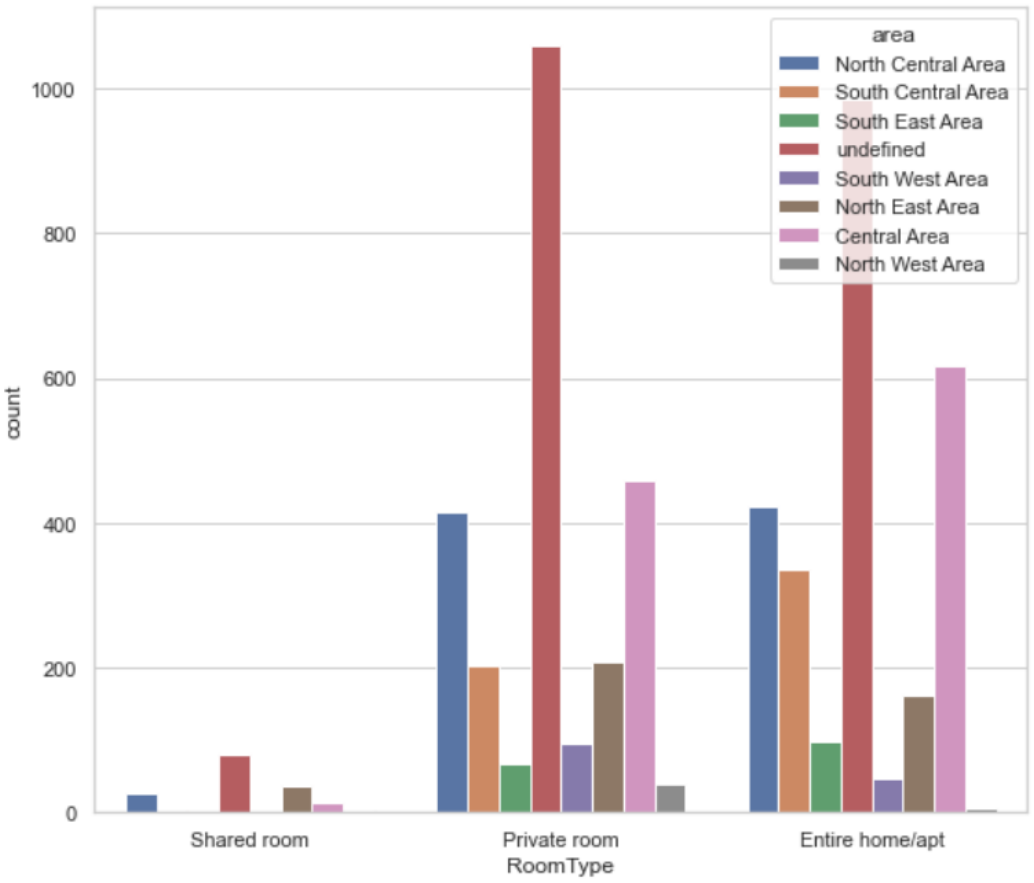


Figure 34. *Dataset Airbnb Merged Datasets - Patterns and Trends IV*

5.5 Dataset 6: Rate

5.5.1 Data Cleaning

- Unnamed Columns

The dataset presented had undefined columns, as seen in Figure 35. To avoid any confusion and to make the experience with this dataset more user-friendly, it is clear from Figure 36 that once these columns were renamed according to the information provided in the dataset, it is possible to identify their properties more quickly.

	THIS IS A SAMPLE DATASET. PLEASE SUBSCRIBE TO YOUR MARKET TO GAIN FULL ACCESS TO ITS DATA!	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	...	Unnamed: 36
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN
1	Listing URL	Property Type	Latitude	Longitude	Star Rating	Number of Active Days	Bedrooms	Has pool	Cleaning Fee	Extra Guest Fee	...	Revenue (2021-01)
2	http://airbnb.com/rooms/47311	entire_home	50.81523	-0.10806	4.3	274	1	f	37	0	...	62
3	http://airbnb.com/rooms/74819	Entire Home	50.82145	-0.14257	4.9	243	3	f	52	0	...	7078
4	http://airbnb.com/rooms/76190	entire_home	50.82369	-0.15587	4.5	365	3	f	63	5.04667	...	5920

5 rows × 46 columns

Figure 35. *Dataset Rate - Unnamed Columns I*

```
rate.columns = [
    "Listing URL",
    "Property Type",
    "Latitude",
    "Longitude",
    "Star Rating",
    "Number of Active Days",
    "Bedrooms",
    "Has pool",
    "Cleaning Fee",
    "Extra Guest Fee",
    "Daily Rate (2020-11)",
    "Daily Rate (2020-12)",
    "Daily Rate (2021-01)",
    "Daily Rate (2021-02)",
    "Daily Rate (2021-03)",
    "Daily Rate (2021-04)",
    "Daily Rate (2021-05)",
    "Daily Rate (2021-06)",
    "Daily Rate (2021-07)",
    "Daily Rate (2021-08)",
    "Daily Rate (2021-09)",
    "Daily Rate (2021-10)",
    "Occupancy Rate (2020-11)",
    "Occupancy Rate (2020-12)",
    "Occupancy Rate (2021-01)",
    "Occupancy Rate (2021-02)",
```

Figure 36. *Dataset Rate - Unnamed Columns II*

- Drop Columns

After analysing the data available in these columns, it was clear that the information was not relevant to occupancy prediction. Therefore, as seen in Figure 37, the method .drop() was used to delete these columns from the dataset.

```
rate.drop(['Listing URL',
           'Property Type',
           'Has pool'],
          axis=1, inplace = True)
```

Figure 37. *Dataset Rate - Dropping Columns*

- Drop Rows

The first two rows of this dataset contained irregular information. According to Figure 38, the first row only had null values, and in fact, the second row contained the column names. The method .drop() was used for removing these rows and guaranteeing that the dataset only contains suitable data for prediction models.

0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1	Listing URL	Property Type	Latitude	Longitude	Star Rating	Number of Active Days	Bedrooms	Has pool	C
2	http://airbnb.com/rooms/47311	entire_home	50.81523	-0.10806	4.3	274	1	f	
3	http://airbnb.com/rooms/74819	Entire Home	50.82145	-0.14257	4.9	243	3	f	
4	http://airbnb.com/rooms/76190	entire_home	50.82369	-0.15587	4.5	365	3	f	

5 rows × 46 columns

Figure 38. *Dataset Rate - Rows Dropped*

- Null Values

Two columns had a total of 23 null values that needed to be processed. As seen in Figure 39, these values were replaced with 0 by using the function.fillna(). The results can be verified once the rate.isnull(). sum() function is called.

```
rate.fillna({'Bedrooms': '0'}, inplace = True)
rate.fillna({'Cleaning Fee':0}, inplace=True)
rate.fillna({'Extra Guest Fee':0}, inplace=True)
```

Figure 39. *Dataset Rate - Null Values*

- Categorical and Numerical Data

The memory usage of a Categorical is proportional to the number of categories plus the length of the data [19]. Considering the number of observations present in this dataset, separating the columns according to their data types would be the best solution to avoid memory shortage.

In Figure 40, we can see that the majority of the columns are in fact numerical, and that both types of data have been sorted.

```
categorical_col = ['Listing URL','Property Type', 'Has pool']

numeric_col = ['Latitude',
               'Longitude',
               'Star Rating',
               'Number of Active Days',
               'Bedrooms',
               'Cleaning Fee',
               'Extra Guest Fee',
               'Daily Rate (2020-11)',
               'Daily Rate (2020-12)',
               'Daily Rate (2021-01)',
               'Daily Rate (2021-02)',
               'Daily Rate (2021-03)',
                'Daily Rate (2021-04)',
                'Daily Rate (2021-05)',
                'Daily Rate (2021-06)',
                'Daily Rate (2021-07)',
                'Daily Rate (2021-08)',
                'Daily Rate (2021-09)',
                'Daily Rate (2021-10)',
                'Occupancy Rate (2020-11)',
```

Figure 40. *Dataset Rate - Categorical and Numerical Data*

5.5.2 Patterns and Trends

- Correlation Matrix

Figure 41 shows that there are high correlations between the following variables:

- The daily_rate_mean and revenue_mean (0.78)
- The daily_rate_mean and the number of bedrooms (0.61)
- The cleaning free and the number of bedrooms (0.56)
- The revenue_mean and the occupancy_mean (0.51)
- The revenue_mean and the number of bedrooms (0.48)

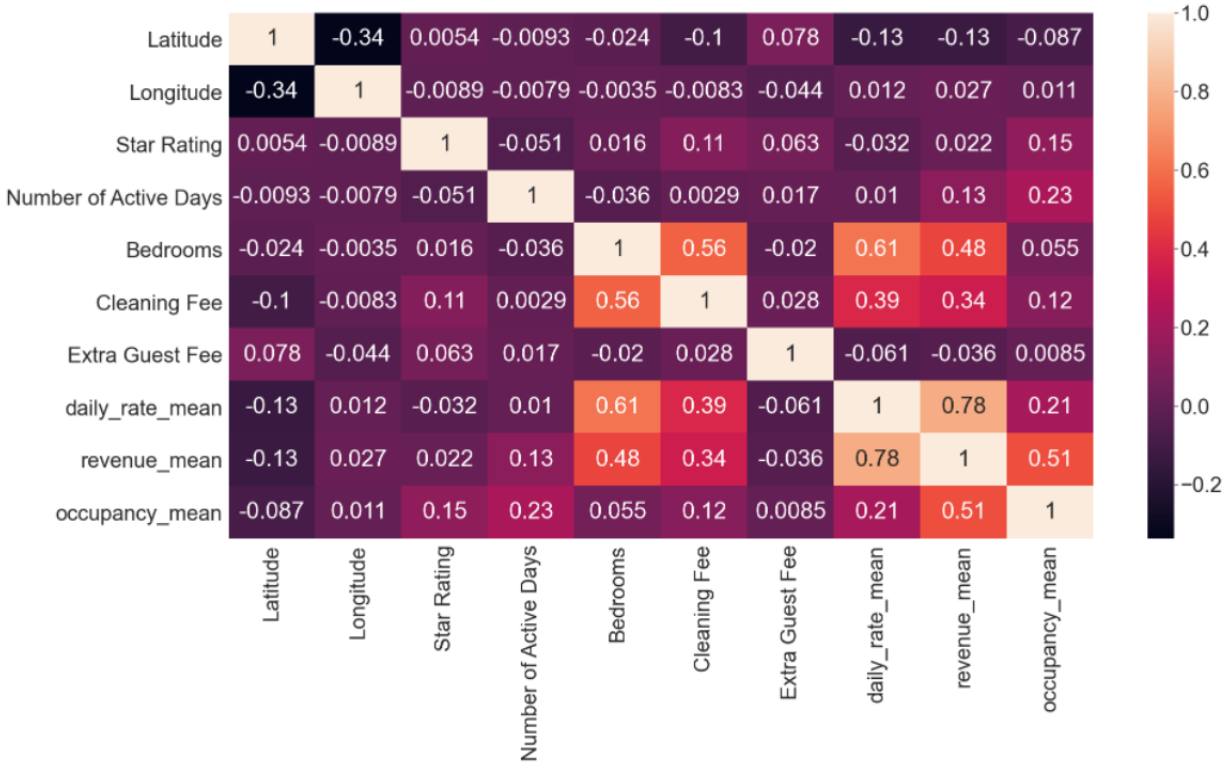


Figure 41. *Dataset Rate - Patterns and Trends*

6. Modelling

This phase is the fourth of the CRISP-DM process and consists of the selection of the appropriate modelling technique [20]. Since this project consists of analysing and combining six datasets, multiple modelling techniques were tested to accuracy of the predictions: Linear Regression, Multiple Linear Regression, K-Nearest Neighbour, Lasso Regression and a sentiment analysis using Natural Language Processing involving Tokenization and the objective is to predict price, occupancy rate and a sentiment analysis based on the reviews.

For the occupancy rate and the price prediction, several modelling techniques are applied to assess the most suitable choice. Tasks that are performed as part of this process include splitting data into training and test sets (datasets were sampled and split into multiple ratios of train and testing data), building the model, fitting and training the mode, performing prediction and comparing results.

These samples were implemented into the modelling techniques previously mentioned to evaluate their performance metrics in R^2, that is statistical measure of the proportion of variability in the predicted variable explained by the regression mode [21]:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2}$$

The Root Mean Square error (RMSE) that measures the standard deviation of the residuals about the fitted regression line [22]:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_i - \hat{y})^2}$$

The Mean Squared Error (MSE) is the mean or average of the square of the difference between actual and estimated values, calculated by the square of the difference between the predicted and actual target variables, divided by the number of data points [23]:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

and MAE, that measures the average magnitude of the errors in a set of predictions, it is the absolute differences between prediction and actual observation where all individual differences have equal weight [24]:

$$MAE = \frac{\sum_{i=1}^{n}|y_{pred,i} - y_i|}{n}$$

6.1 Linear Regression

This modelling technique was used for the Listing (dataset 1) and Rate (dataset 6) to help predict price and occupancy rate, respectively. This type of analysis estimates the coefficients of the linear equation, involving one or more independent variables, that best predict the value of the dependent variable [25].

A linear regression line has an equation of the form $\boldsymbol{Y = a + bX}$, where $\boldsymbol{X}$ is the explanatory variable and $\boldsymbol{Y}$ is the dependent variable. The slope of the line is $\boldsymbol{b}$, and $\boldsymbol{a}$ is the intercept (the value of $\boldsymbol{y}$ when $\boldsymbol{x} = 0$) [26].

6.1.1 Price Prediction

Two variables were chosen 'price' (X) and 'availability_365' (Y); train and test ratio was 60/40; the test set generated a Mean Absolute Error of 89.8249752621703, Mean Squared Error of 13490.4104221186 and Root Mean Squared Error of 116.14822608253.

6.1.2 Occupancy Rate Prediction

Two subsets of the dataset "Rate", the first using 'occupancy_mean' and the second one using 'Occupancy Rate (2020-11)'; train and test ratio was 80/20 and 70/30, respectively; the first test set generated a R2 of 0.287225, meaning low accuracy and the second test set generated a score of 0.936513 represented on Figure 42.

Figure 42. *Occupancy Rate Linear Regression*

6.2 Multiple Linear Regression

Multiple linear regression is used to estimate the relationship between two or more independent variables and one dependent variable. This modelling technique was used for the Listing (dataset 1) in an attempt to improve the prediction of the price [27].

6.2.1 Price Prediction

Four variables were chosen: 'price' (Y) and ''calculated_host_listings_count', 'room_type_Cat' and 'city_Cat' (X) and the prediction can be visualised on Figure 43. The results generated were Root Mean Squared Error: 14049.0189534251, Mean Absolute Error: 386.0298015715187, Mean Squared Error: 197374933.5536991 and R2 score: 0.000229925624612215.

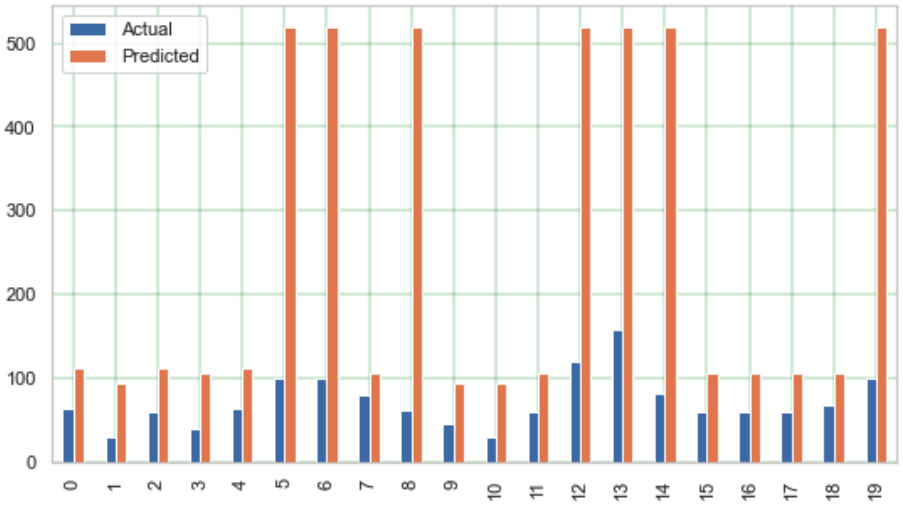


Figure 43. *Multiple Linear Regression - Price Prediction on Listing*

6.3 K-Nearest Neighbour

6.3.1 Price Prediction

A list of conditions created based on the price column and the values were named ‘economic’, ‘low-mid’, ‘high-mid’ and ‘high’ ranging from 100 dollars to 600 dollars and the algorithm was tested using the ‘price_rng_Cat’; train and test ratio was 60/40; the model resulted of an accuracy of 91% at K=3 and at K=70 the score is 86%, this means that small K values are not suitable. This is the best performing model for price prediction as observed on Figure 44.

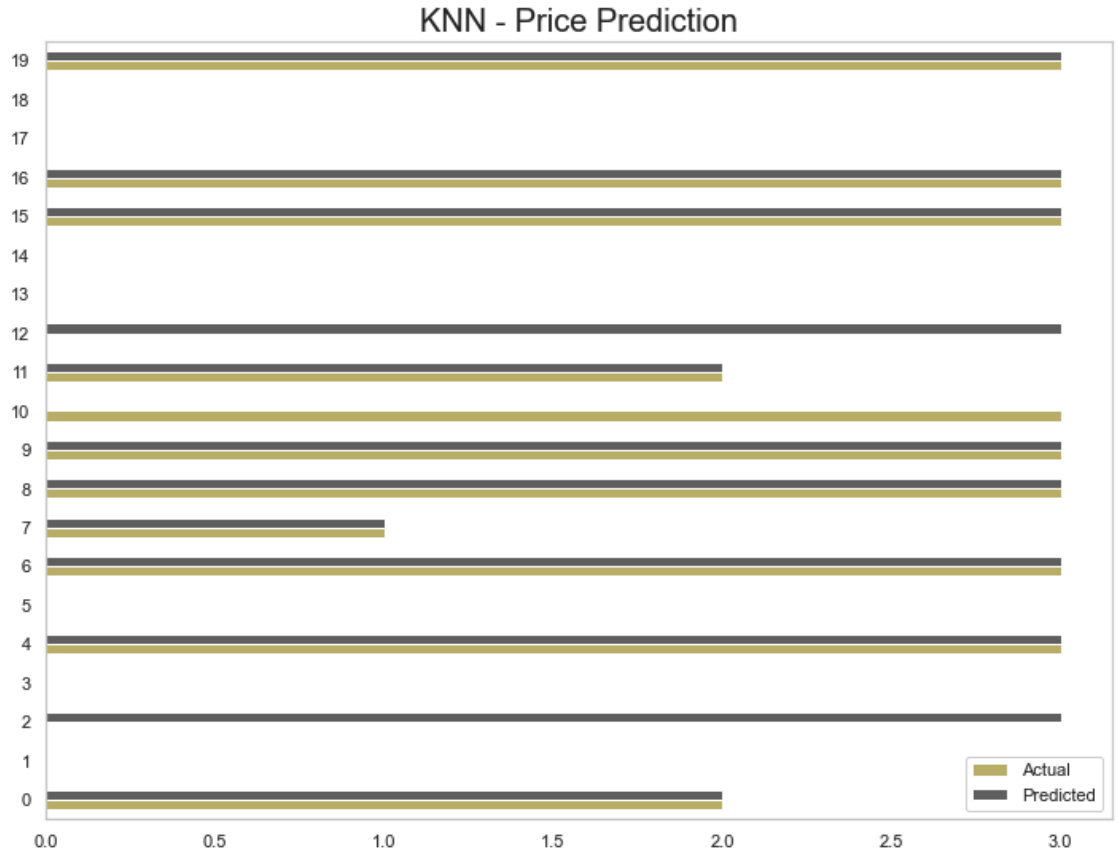


Figure 44. *KNN Price Prediction - Listing*

6.4 Lasso Regression

The Lasso Regression technique, also known as Penalised Regression, is based on simple models with fewer parameters. It eliminates the overfitting issues that linear regression previously had and delivers better prediction accuracy than other regression models [28].

6.4.1 Occupancy Rate Prediction

Based on the occupancy of the month of November "Occupancy Rate (2020-11)", the additional parameters were alpha (set to 3.5, identified via a function, see Figure 45), max_iter = 100 (number of iterations) and tol = 0.1, the test set generated a score of 0.93656.

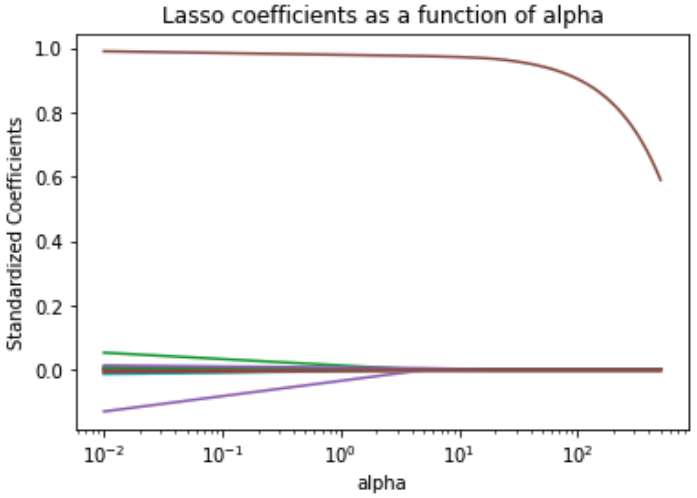


Figure 45. *Lasso Coefficient as a function of alpha*

6.5 Ridge Regression

Ridge Regression is also known as Tikhonov Regularisation, the model is set up with all variables given; it adds a factor of sum of squares of coefficients in the optimization objective but does not remove variables with low relationships from the model, it brings the coefficients of these variables closer to zero. This model solves a regression model where the loss function is the linear least squares function and regularisation is given by the l2-norm [29].

6.5.1 Rate Prediction

The technique was tested with the dataset Rate (dataset 6). The results generated were Root Mean Squared Error of 83.9987249357209, Mean Absolute Error of 1.799879, Mean Squared Error: 83.9987249357209 and R2 score: 0.936499.

6.6 Natural Language Processing (NPL) : Sentiment Analysis - Tokenization

The dataset “Review” contains more than 211 thousand observations, for this reason, in order to optimise computing time, a sample of 2000 observations were selected to perform the analysis. The sentiment analysis uses the words from the ‘comments’ and the technique applied to preprocess it was word embeddings.

The categorical data was treated using the factorise() method to convert the values from categorical to numeric, the next step is to tokenize the words with the help of Tokenizer. Tokenization is the process of converting text into tokens before transforming it into vectors. This analysis is done by tokenizing the reviews into words.

The architecture of our model consists of an embedding layer, an LSTM layer, and a Dense layer at the end and to avoid overfitting Dropout mechanism in-between the LSTM layers has been introduced.

The Figure 46 shows the section where it shows the train of the sentiment analysis model for 5 epochs on the whole dataset with a batch size of 32 and a validation split of 20%. The model reached a score of 98% in the test set and 99% on the train set.

```
: history = model.fit(padded_sequence,sentiment_label[0],validation_split=0.2, epochs=5, batch_size=32)

  Train on 1543 samples, validate on 386 samples
  Epoch 1/5
  1543/1543 [==============================] - 31s 20ms/sample - loss: 0.3193 - acc: 0.9741 - val_loss: 0.0750 - val_ac
  c: 0.9896
  Epoch 2/5
  1543/1543 [==============================] - 29s 19ms/sample - loss: 0.0369 - acc: 0.9948 - val_loss: 0.0659 - val_ac
  c: 0.9896
  Epoch 3/5
  1543/1543 [==============================] - 29s 19ms/sample - loss: 0.0325 - acc: 0.9948 - val_loss: 0.0616 - val_ac
  c: 0.9896
  Epoch 4/5
  1543/1543 [==============================] - 28s 18ms/sample - loss: 0.0321 - acc: 0.9948 - val_loss: 0.0602 - val_ac
  c: 0.9896
  Epoch 5/5
  1543/1543 [==============================] - 29s 19ms/sample - loss: 0.0319 - acc: 0.9948 - val_loss: 0.0580 - val_ac
  c: 0.9896
```

Figure 46. *Sentiment Analysis - Model Building*

7. Evaluation

7.1 Results Evaluation

Changes to the ratio size, which is the first advice, did not result in any substantial improvement in the accuracy of the Regression models. In this instance, Linear and Multiple produced similarly poor results where feature selection and scaling were equally unsatisfactory as observed on table 1.

Another technique was used to improve and more precisely predict the price: K-Nearest Neighbour on the Listing dataset. Tests showed promising results when K was initialised with a random number, but lower values of K resulted in more unstable decision boundaries. There are no predefined statistical methods for determining the most favourable value of K [30], though it has been suggested that the best value for K is the square root of N. The best K value was found by the root square of the total number of samples (N) as shown in table 1, and the model yielded an accuracy of 85 percent with K=83.

Price - Listing								
Model	Ratio	Intercept	Coefficient	MAE	R2	RMSE	MSE	Score
Linear Regression	6:4	64.50394809	0.00010087	89.82497526	-0.0003330123073	116.1482261	13490.41042	NA
Multiple Linear Regression	7:3	524.7107988	-5.83326112e-02 -1.01995836e+14 1.01995836e+14 -5.59082031e+00	386.0298016	0.0002299256246	14049.01895	197374933.6	NA
K-Nearest Neighbour	6:4	NA	NA	0.3572196345	NA	0.9974887853	0.9949838767	0.8559656037

Table 1. *Price Prediction Results*

The occupancy rate was predicted using similar methods. Starting with Linear Regression, we were able to build a better forecast by picking one month; both Lasso and Ridge regression provided an R2 of 0.93; the greater the R2 score, the better the model's accuracy, as shown in table 2.

Occupancy Rate					
Model	Ratio	MAE	R2	RMSE	MSE
Linear Regression - Mean	8:2	19.638265	0.287225	25.673476	659.1273887
Linear Regression - Occupancy 11	7:3	1.804119	0.936513	9.164127	83.98122852
Lasso Regresion	NA	1.621983	0.93656	9.160698	83.91838838
Ridge Regression	NA	1.799879	0.936499	9.165082	83.99872494

Table 2*. Occupancy Rate Prediction*

8. Deployment

The deployment of the application follows the aforementioned phases with the objective of defining where the use of the data preparation and models will be used as a resource. The majority of studies and research on CRISP framework methodology do not include any specific following requirements for this last phase as its goal is to describe the actions aimed at supporting development or further steps.

The integration of this last phase into a production environment intends for the application to be presented to the user. In our scenario, it aims to show the data behind the models through visualisations and maps, as well as supporting the application's development and directing previous phases to a final product. According to The CRISP-DM Model [31], this phase might vary from a basic report to a comprehensive application, such as in our instance. This phase is divided into the following steps:

- Plan Deployment
- Plan Monitoring
- Plan Maintenance
- Final report
- Review report

Planning Deployment of an application is critical since it demands following the previous phases in order to develop a solution that users find straightforward and comfortable to use [32]. Sketching an abstract concept of this idea is feasible with Figma, a powerful design tool that employs graphics objects to build digital products such as web applications [33]. Furthermore, these applications outperform software applications being used in organisations due to their benefits, such as operation speed, scalability, and affordability.

Following the completion of the layout, frameworks are the next phase in web development. [34] Django is a popular framework that allows Python to be used as the back end of a web application and integrates with the React library to produce applications that can be expanded and adapted according to customer demands. [35] These tools are represented on Figure 47.

Figure 47. *Resources for Plan Deployment.*

Plan Monitoring and Maintenance has a great importance for this phase especially when the production environment comes to an end. Once the software is fully deployed to its full capability, errors may arise during or after deployment, preventing the application from functioning as planned. Not only can cloud platforms host a web application, but they may also provide monitoring and maintenance as needed.

Firebase is one of the two platforms used for this application. It hosts the front end and provides services such as monitoring health, testing code, and application compilation [36]. It also generates reporting, analytics, and insights with the help of Google Analytics.

Heroku is the platform in charge of hosting the back end of the application, where the previously mentioned data models will be used, from displaying property locations to containing data for visualisations of these models' and where the API behind the application will be hosted [37]. By using these platforms, they have the potential to boost the application's performance and bring benefits, such as pay as you go and scaling production to support usage.

When it comes to application maintenance, MongoDB is the most suitable choice: it is a NonSQL Database that consists of measures to prevent the data from being improperly handled since it is the major source behind the application [38]. The features available on this database range from document-oriented to indexing and offering advantages when deciding

on implementation. However, the replication of sets in the event that data is unavailable and load balancing to scale as needed are where it will ensure that the application runs and models learn with the use of the final product. These tools are represented on Figure 48.

Figure 48. *Resources for Monitoring and Maintenance.*

To display the predictions, Mapbox and Chart.js were used. Mapbox is an open source map API that is used on a large scale by companies such as Uber and Snapchat [39], while Chart.js is an JavaScript library that handles data easily to display visualisation that can be fully modified and fast responsive [40].

Final Report for the application shows the development of the project through a summary of commits through Github, Team Agenda and Basecamp. These commits are results of experiences and activities that have occurred to guarantee the presentation of the final product the web application named before submission Homely.

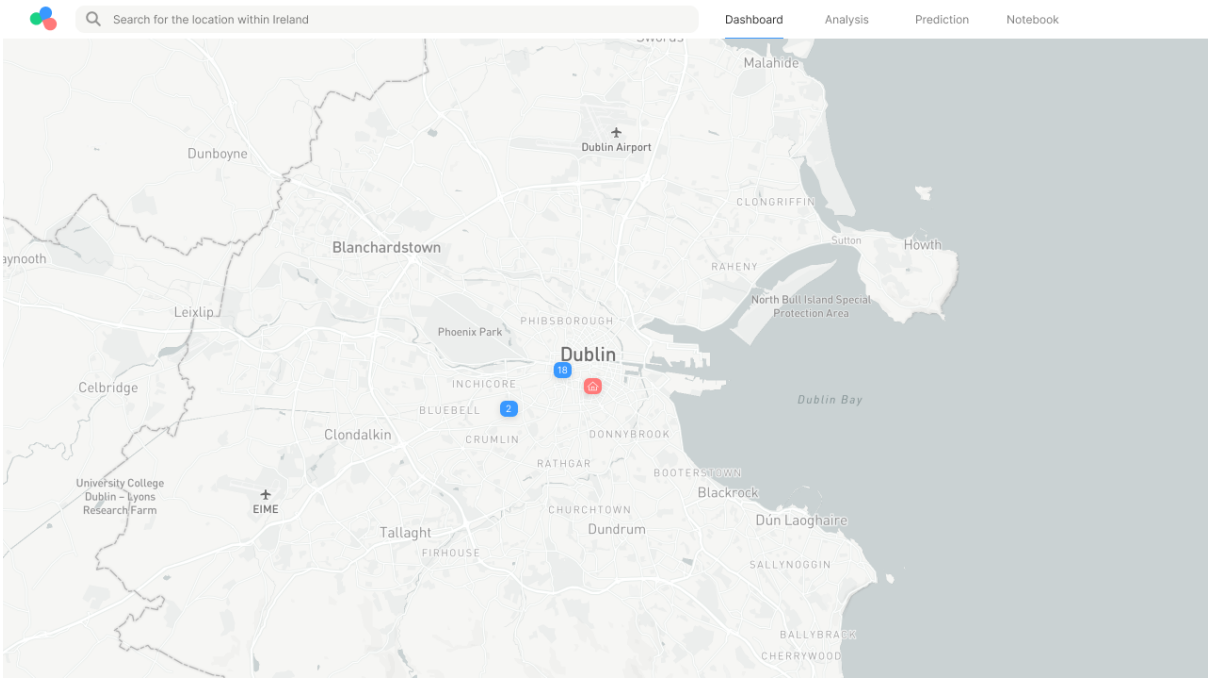


Figure 49. *Figma Dashboard*

In regards to the Review Report, this project demonstrates that the application has potential for feature improvements: because it is essentially a prototype, and thus may accomplish the first project concept with more research on data manipulation, data modelling, and web development. Misleading approaches that have taken a part of the development from the preparation, modelling and web development did result in learning outcomes time guaranteeing experience for future projects. The final dashboard layout is presented on Figure 49.

9. Conclusion

Through this report, we have discussed the primary objectives of this project from a business perspective, where we also approached the possible competitors and the differences between their features.

By implementing CRISP framework into this research we have also successfully achieved the data mining goals, such as analysing and evaluating property related data in Ireland. Following the implementation of modelling techniques, the best methods were identified, as well as the outcomes evaluated and compared. Each dataset's patterns and trends were displayed, and we explored how these may affect the predictions.

The deployment phase began with the introduction of a prototype using Web Development frameworks. The prototype illustrates the final dashboard layout, which includes the functionalities that will be accessible once the application is completely operational. The completion of this phase greatly contributed to the project's credibility and confirmed its viability.

10. References

Introduction

[1] Sanford, W. and DuBois, D., 2022. COVID-19 impact on hotels and short-term rentals. [online] STR. Available at: <https://str.com/whitepaper/covid-19-impact-on-hotels-and-short-term-rentals-airdna> [Accessed 15 May 2022].

[2] Airbnb. 2022. About us. [online] Available at: <https://news.airbnb.com/about-us/> [Accessed 15 May 2022].

[3] Schmelzer, R., 2022. Data Science vs. Machine Learning vs. AI: How They Work Together. [online] SearchBusinessAnalytics. Available at: <https://www.techtarget.com/searchbusinessanalytics/feature/Data-science-vs-machine-learning-vs-AI-How-they-work-together> [Accessed 15 May 2022].

Research Methodology

[4] Luna, Z. (2021). Understanding CRISP-DM and its importance in Data Science projects. [online] Analytics Vidhya. Available at: https://medium.com/analytics-vidhya/understanding-crisp-dm-and-its-importance-in-data-science-projects-91c8742c9f9b. [Accessed 01 May 2022]

[5] Smart Vision - Europe. (2017). Building and Applying Predictive Models in IBM SPSS Modeler training webinar. [online] Available at: https://www.sv-europe.com/crisp-dm-

methodology/. [Accessed 15 May 2022]

Business Understanding

[6] Medium. 2022. 5 Ways to Apply Data Science to Real Estate. [online] Available at: <https://towardsdatascience.com/5-ways-to-apply-data-science-to-real-estate-e18cdcd0c1a6 > [Accessed 06 March 2022].

[7] Georgina Guthrie, (2022). 'What is Situation Analysis, and why is it so important?'. [online]. Blog, Cacoo. Available at:https://cacoo.com/blog/what-is-situation-analysis-and-why-is-it-so-important/ [Accessed 07 May 2022].

Data Understanding

[8] Wickramasinghe, S., 2022. Jupyter Notebooks for Data Analytics: A Beginner's Guide. [online] BMC Blogs. Available at: <https://www.bmc.com/blogs/installing-jupyter-for-big-data-and-analytics/> [Accessed 15 May 2022].

[9] Docs.jupyter.org. 2022. Try Jupyter — Jupyter Documentation 4.1.1 alpha documentation. [online] Available at: <https://docs.jupyter.org/en/latest/start/index.html> [Accessed 15 May 2022].

[10] Insideairbnb.com. 2022. [online] Available at: <http://insideairbnb.com/dublin> [Accessed 15 May 2022].

[11] Public.opendatasoft.com. 2022. Airbnb listings. [online] Available at: <https://public.opendatasoft.com/explore/dataset/air-bnb-listings/table/?disjunctive.neighbourhood&disjunctive.column_10&disjunctive.city&q=dublin> [Accessed 15 May 2022].

[12] App.airbtics.com. 2022. Airbnb analytics for Short-term rental buyers and Hosts. [online] Available at: <https://app.airbtics.com/airbnb-data/ireland/0/dublin> [Accessed 15 May 2022].

Data Preparation

[13] ThoughtCo. (n.d.). Detect the Presence of Outliers With the Interquartile Range Rule. [online] Available at: https://www.thoughtco.com/what-is-the-interquartile-range-rule-3126244#:~:text=Using%20the%20Interquartile%20Rule%20to%20Find%20Outliers&text=Calculate%20the%20interquartile%20range%20for [Accessed 05 May 2022].

[14] Jakevdp.github.io. (n.d.). Handling Missing Data | Python Data Science Handbook. [online] Available at: https://jakevdp.github.io/PythonDataScienceHandbook/03.04-missing-values.html. [Accessed 06 May 2022]

[15] Pandas.pydata.org. (n.d.). pandas.cut — Pandas 1.4.2 documentation. [online] Available at: https://pandas.pydata.org/docs/reference/api/pandas.cut.html [Accessed 06 May 2022].

[16] GeeksforGeeks. (2018). Generating Word Cloud in Python. [online] Available at: https://www.geeksforgeeks.org/generating-word-cloud-python/. [Accessed 13 May 2022]

[17] GeeksforGeeks. (2019). Convert the column type from string to datetime format in Pandas dataframe. [online] Available at: https://www.geeksforgeeks.org/convert-the-column-type-from-string-to-datetime-format-in-pandas-dataframe/ [Accessed 6 May 2022]

[18] GeeksforGeeks. (2018). Median() function in Python statistics module. [online] Available at: https://www.geeksforgeeks.org/python-statistics-median/. [Accessed 13 May 2022]

[19] Pandas.pydata.org. (n.d.). Categorical data — pandas 1.4.2 documentation. [online] Available at: https://pandas.pydata.org/docs/user_guide/categorical.html#categorical-memory [Accessed 13 May 2022].

Modelling

[20] Sridharan, M. and Sridharan, M., 2022. CRISP-DM: A Framework For Data Mining & Analysis. [online] Think Insights. Available at: <https://thinkinsights.net/digital/crisp-dm/#Modeling> [Accessed 15 May 2022].

[21] Science, D. and Analysis, H., 2022. How to interpret R-squared in regression analysis?. [online] Knowledgehut.com. Available at:

<https://www.knowledgehut.com/blog/data-science/interpret-r-squared-and-goodness-fit-regression-analysis> [Accessed 11 May 2022].

[22] Statistics How To. 2022. RMSE: Root Mean Square Error. [online] Available at: <https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/#:~:text=Root%20Mean%20Square%20Error%20(RMSE)%20is%20the%20standard%20deviation%20of,the%20line%20of%20best%20fit.> [Accessed 11 May 2022].

[23] Python Pool. 2022. How To Calculate Mean Squared Error In Python. [online] Available at: <https://www.pythonpool.com/mean-squared-error-python/> [Accessed 11 May 2022].

[24] Medium. 2022. What are RMSE and MAE?. [online] Available at: <https://towardsdatascience.com/what-are-rmse-and-mae-e405ce230383> [Accessed 11 May 2022].

[25] Statistics Solutions. 2022. What is Linear Regression? - Statistics Solutions. [online] Available at: <https://www.statisticssolutions.com/free-resources/directory-of-statistical-analyses/what-is-linear-regression/> [Accessed 09 May 2022].

[26] Stat.yale.edu. 2022. Linear Regression. [online] Available at: <http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm#:~:text=A%20linear%20regression%20line%20has,y%20when%20x%20%3D%200).> [Accessed 09 May 2022].

[27] Jmp.com. 2022. Multiple Linear Regression. [online] Available at: <https://www.jmp.com/en_au/statistics-knowledge-portal/what-is-multiple-regression.html> [Accessed 09 May 2022].

[28] scikit-learn. 2022. sklearn.linear_model.Lasso. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html> [Accessed 09 May 2022].

[29] Medium. 2022. Appropriating Tikhonov Regularization (Ridge Regression). [online] Available at: <https://medium.com/analytics-vidhya/appropriating-tikhonov-regularization-ridge-regression-c91680b66dfc> [Accessed 09 May 2022].

Evaluation

[30] Band, A. (2020). How to find the optimal value of K in KNN? [online] Medium. Available at: https://towardsdatascience.com/how-to-find-the-optimal-value-of-k-in-knn-35d936e554eb. [Accessed 11 May 2022]

Deployment

[31] Shearer, C. (2000). The CRISP-DM Model: The New Blueprint for Data Mining. Journal of Data Warehousing. [PDF]. Available at: https://pdfslide.net/documents/the-crisp-dm-model-the-new-blueprint-for-data-mining-shearer-colin.html [Accessed 10 May 2022].

[32] Brown, M. (2016). Phase 6 of the CRISP-DM Process Model: Deployment. Dummies. Available at: https://www.dummies.com/article/technology/information-technology/data-science/general-data-science/phase-6-of-the-crisp-dm-process-model-deployment-148174/ [Accessed 09 May 2022].

[33] Figma. (2022). Creative tools meet the internet. Available at: https://www.figma.com/about/ [Accessed 11 May 2022].

[34] Goel, A. (2022). 10 Best Web Development Frameworks. Hackr.io. Available at: https://hackr.io/blog/web-development-frameworks [Accessed 10 May 2022].

[35] Brewster, C. (2022). 9 Examples of Companies Using Django in 2022. Trio. Available at: https://www.trio.dev/blog/django-applications [Accessed 10 May 2022].

[36] Mehta, A. (2021). Firebase for Startups: A Must-Have or Non-Essential. Appinventiv. Available at: https://appinventiv.com/blog/firebase-for-startups/ [Accessed 10 May 2022].

[37] Heroku (2020). Cloud Application Platform | Heroku. [online] Heroku.com. Available at: https://www.heroku.com/.

[38] Taylor, D. (2022). What is MongoDB? Introduction, Architecture, Features & Example. Guru99. Available at: https://www.guru99.com/what-is-mongodb.html [Accessed 10 May 2022].

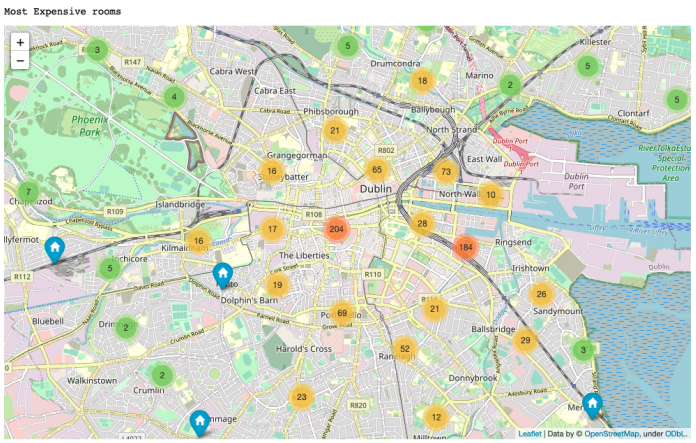
[39] Mapbox. (2022). Mapbox Our Vision. Available at: https://www.mapbox.com/about/company [Accessed 10 May 2022].

[40] Tobiasahlin.com. (n.d.). Data visualization with Chart.js: An introduction. [online] Available at: https://tobiasahlin.com/blog/introduction-to-chartjs/ [Accessed 11 May 2022].

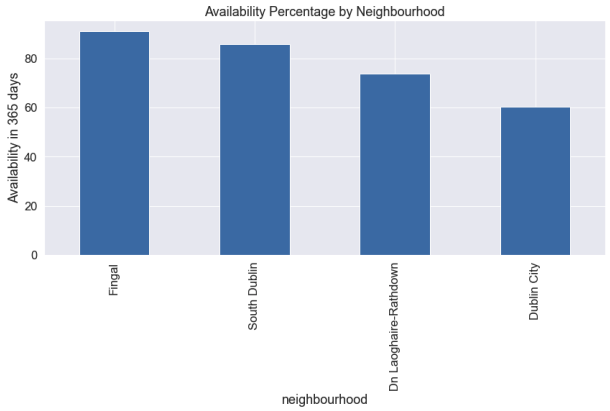
11. Appendices

11.1 Additional Findings

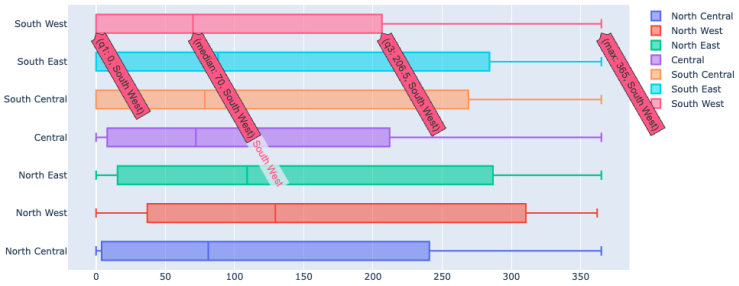
11.1.1 Most Expensive Rooms



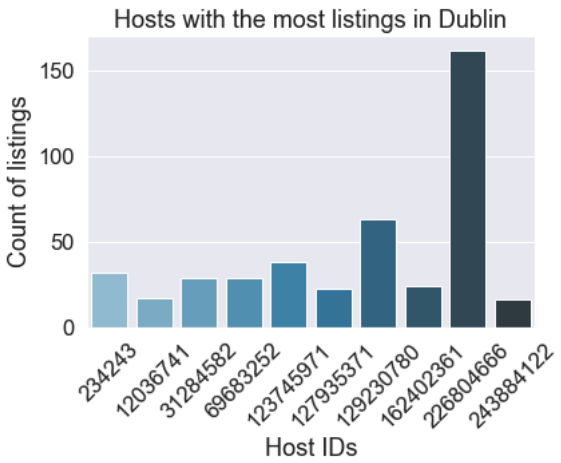
11.1.2 Availability Percentage by Neighbourhood



11.1.3 Availability by Area



11.1.4 Hosts with most listings



11.1.5 Density and distribution of prices for each neighbourhood.

11.1.6 Decision Tree - Review

```
dt = DecisionTreeClassifier(random_state=0)
dt.fit(X_train,y_train)
y_pred_test = dt.predict(X_test)
print("Training Accuracy score: "+str(round(accuracy_score(y_train,dt.predict(X_train)),4)))
print("Testing Accuracy score: "+str(round(accuracy_score(y_test,dt.predict(X_test)),4)))
```

```
Training Accuracy score: 0.9994
Testing Accuracy score: 0.685
```

11.2 Reflective Journal

11.2.1 Giovanni Andrade

Joining a new group for the final year project is always a new learning outcome to me that I am always happy to face, especially when it comes to meeting people that I have not met through the course before. As the last year of studies, bringing ideas to the board of what sort of application we could develop was very intuitive as we reached an idea of the current application.

From there the first move was the beginning of the research for the application datasets, the first research on my side was the discovery of a dataset bank from the Irish government that has information about all council data from house prices to amenities. However, finding that our first project idea had great potential it would take longer to develop before the submissions date, so by working with the airbnb datasets found by Luciana we had the scope of the project for short term rental.

During the work of the airbnb datasets sections were assigned to each member based on choice and strong qualities, after Luciana's introduction to the Python notebook, the datasets rate and review datasets assigned to me. For both datasets my responsibility was to create a prediction model with an accuracy goal of 80% to 95% during the test and trains and that these models would give the chance to create visualisations for the dashboard.

With those steps done the next step was to start the development of a dashboard as my strong qualities are around development of web applications the dashboard development was in my responsibilities, however with input of the group during the development the dashboard it is our result of our project application.

For the documentation of the application the deployment phase of CRISP-DM is where all the details of the application development is available briefly describing the technologies chosen for the web application. The result of this project made me really excited to get into a software development role as industries use the approaches of our project, which not only is the outcome of learning all these programming languages used in the project very important for my professional career but creating an understanding of real life practice of daily work in a team.

11.2.2 Luciana Teixeira

Firstly, the most obvious thing that I discovered is how important it is to organise and assign tasks when working in a group, while it may seem simple, it is imperative so that the project can be moved forward successfully; by tuning in my own management skills acquired at my current job I set up shared documents, meeting documents and took notes of the tasks and responsibilities and deadlines, I personally believe it is very important to have transparency and visibility when working as part of a group so that the team understands each other's commitments, who is available to help if needed and last but not least, to manage time wisely.

Throughout our collaboration, new ideas were introduced and the scope of the project changed slightly, I learned that it is important to be open minded and take into consideration everyone's ideas, for me it was important to have a mixed gender team from different countries focusing on diversity which I credit a lot of personal and academic growth.

I have completed my Computer Science degree in another institution, so it was challenging to participate in the curriculum having never seen Python before. However, I have completed various online courses to catch up and I learned a lot in the last semester; I have a passion for anything data-related so while it was absolutely challenging it was also very rewarding.

The most difficult part was to find a suitable dataset, so some of my tasks included finding, pre-processing, cleaning and preparing the datasets for this project. I am personally responsible for the preparation of Listings, AirbnbAirbnb_listing 1 and Airbnb_listing 2 and Rate; these tasks took around 40 hours to be completed.

One of the most rewarding parts of working with the dataset was to create beautiful visualisations, I found that to be a very interesting task and I enjoyed learning about many different libraries such as Folium, I am solely responsible for the visualisations on the combined datasets (AirbnbAirbnb_listing 1 and Airbnb_listing 2) and I am partially responsible for the visualisations on Listings; these tasks took about 30 hours to be completed.

Needless to say, the part of the project that took more time was to improve the predictions and testing various modelling techniques to find a suitable one. I mistakenly underestimated the amount of time and effort that could go into reading documentations, watching tutorial videos in order to gain better insight on how to work with the data and how to improve the accuracy of the models. I am personally responsible for the Price Prediction and this task took approximately 40 hours to be completed.

11.2.3 Marcelle Louise

When I joined CCT College in 2021, I did not expect to be completing the course with the amount of both technical and soft skills I have today. Because I completed my bachelor's degree in a different institution, I had no previous knowledge in Data Analysis or Data Exploration. In fact, Python wasn't even introduced in the previous course I completed. However, this year I had the opportunity to completely immerse myself into this computing language. This module in particular had a great impact in improving my knowledge in Python.

By assisting the team with the EDA task for three out of the six datasets (Listings, Review and Rate), I was able to put into practise the multiple steps acquired for data analysis, data preparation, and the differences between the algorithms used for modelling and predictions. Although I struggled to complete this task, my colleagues were supportive and that boosted my confidence to keep learning and trying, even after I encountered errors and was not able to meet the entire task requirements. I also made sure to communicate with them at all times about these struggles, which permitted them to assist me.

One of my main achievements in this project was in regards to the reporting. I am responsible for completing the Abstract, Research Methodology, Conclusion, Data Preparation chapters. The last one mentioned covered the many steps taken to make the data available suitable for modelling algorithms but also approached insights of each datasets after they were cleaned. In addition, I was responsible for revising all chapters and adding changes accordingly once all team members completed their chapters.

Due to my interest in Design, I was responsible for creating the poster layout and content with Figma. The design focused on showing the accomplishments of every chapter, including some of the data exploration findings. Once the layout was completed, Muhammad then assisted on getting the content distributed into the sections. Luciana and Giovanni provided screenshots of the best visualisations to be added to the poster. This task alone took me approximately 5 days to complete since I needed to incorporate new sections and content to match lecturer's feedback.

By completing this project, I realised how often we underestimate our capabilities and capacity to learn. I am confident to say that this research has helped me strengthen my data analysis, preparation, and modelling skills, which will prove useful when I begin to work in the IT field.

11.2.4 Muhammad Shabaz

Student number: 2018092

This report is based on individual reflection as a final year project. Student of Bachelor of Science (Hons) computer and information technology. College of computing and technology Dublin, Ireland.

Planning Phase

The initial phase was to come up with an idea that must be different and feasible to work on. We decided to research the properties data, find 6 different datasets from the Airbnb website and another task was to make an application that will help people in finding the right properties according to their wants and needs. Collected six different datasets and did the in-depth analysis using python on the Jupyter notebook. Find useful information in the form of visualisations, prediction, data modelling, and other data regarding information. Finally, we wrote a detailed research report and an application dashboard using different programming languages.

Acknowledgment

In this report, I am going to share my own experience in working with a team on this final year group project “Financial Risk and Market Trends Through Predictive Data Analysis”. It was a really great experience full of a passionate team, learning different ways to achieve the project goals, and completing deliverables within tight deadlines. We were a team of four students who worked on this project. I would like to sincerely thank you all for the hard work, diligence and self-motivation, without which it would not have been possible. Surely, the success of this project will give us a competitive edge over our other groups.

Individual contribution

1. I started working on this project from data analysis, my task was to clean and explore the datasets. Started by Cleaning and preparing the Calendar.csv dataset and finding some useful information by doing Exploratory data analysis and visualisations. I find some useful visualisation regarding the listing's dataset as well.

2. My second task was research work. I started working on the research report. My task was to write on business logic, and data understanding. Report design, spelling check, and grammatical errors.

3. I have worked on the poster with Marcelle on the poster of our project. We have spent a few days finalising the poster. For the final version, I covered most of the work for the poster, almost 50 percent. I have spent more than two days completing the poster for the final version.

4. Lastly, I have offered my contribution to writing the content for our application dashboard and wrote some content as well. I was also responsible for managing my GitHub account as we all were using my account for a repository. Teachers were also given access to GitHub accounts to see our progress.

Project Outcomes

I have learned many things by working on this project, especially, working in a team that teaches you how to help your team members, and how to work hard to fulfil your team's expectations. Dealing with time management is a big lesson that one can learn from this kind

of environment. Dividing your work according to the deliverables has taught me how to organise your work. My data analytics and python knowledge was limited, spending several days just to figure out and explore a particular data by visualisations was a big challenge for me and now I have excellent hands-on experience using python for data analysis. Utilising different ways to get the right stuff for your research work and proper referencing has developed my research skills.

11.3 Additional Links

11.3.1 Poster

https://www.figma.com/file/FE74jzXGnciJmLpqBc3J8x/Final-Project-CA?node-id=207%3A3

11.3.2 Prototype

https://www.figma.com/file/2GdIL5u1gApyICerxy0kKx/Dashboard

11.3.3 Final Dashboard

https://homelydash.web.app/#

11.3.4 Final Code

https://github.com/Shahbaz906/finalyearproject