**CCT College Dublin**

**Assessment Cover Page**

|  |  |
| --- | --- |
| **Module Title:** | *Programming for DA*  *Stats for Data Analytics*  *ML for Data Analysis*  *Data Prep & Vis* |
| **Assessment Title:** | *MSC\_DA\_CA1* |
| **Lecturer Name:** | *Taufique Ahmed/Cristiani Eccher*  *Kayoum Khbuli*  *Muhammad Iqbal*  *David McQuaid* |
| **Student Full Name:** | Erick Eduardo Rios Trevino |
| **Student Number:** | 2024488 |
| **Assessment Due Date:** | *8th November 2024* |
| **Date of Submission:** | *6th November 2024* |

**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. |

## Abstract

This project aims to study the Tourism industry in Ireland, paying particular attention to regional accommodation, activities, and attractions. Various statistical techniques were employed to explore and analyse the data, such as investigation of central tendencies and the suitability of various statistical distributions to describe the data. Alongside statistical analysis, various machine learning techniques (linear regression, random forest, K-mean etc.) were used to predict the relationship between the three features of interest: accommodation, activities, and attractions. This project found that the dataset was best described by a Poisson Distribution. Linear regression was the machine learning approach most successful at predicting the availability of accommodation given the proximity of activities/attractions with an R2 value of 96%. Numerous data visualizations -including scatter plots, histograms, heatmap and interactive geographical maps- were produced and are provided in this report to best represent this data in a clear and coherent manner.

## Introduction

In this project, I explored three datasets to uncover insights about accommodations available to tourist in Ireland. Key questions to answer include:

* How many accommodations, activities and attractions are available per county?
* What is the average number of accommodations, activities and attractions per county?
* Is there any relationship amongst accommodation, activities, and attractions?

Additionally, I analysed the maximum and minimum values for each feature to identify the most and least crowded counties and zones.

Lastly, I used machine learning models in order to predict accommodation availability based on the number of nearby activities and attractions. Through these analyses, this project aims to provide valuable insights into the distribution of tourist resources across Ireland.

Contents

[Abstract 2](#_Toc181727858)

[Introduction 2](#_Toc181727859)

[1. Exploratory Data Analysis 4](#_Toc181727860)

[1.1. Irish Accommodation Dataset 4](#_Toc181727861)

[1.2. Irish Attractions Dataset 5](#_Toc181727862)

[1.3. Irish Activities Dataset 6](#_Toc181727863)

[1.4. Creating new one Dataset 7](#_Toc181727864)

[1.4.1. EDA Conclusion 8](#_Toc181727865)

[2. Visualization 8](#_Toc181727866)

[2.1. Pair plot 8](#_Toc181727867)

[2.2. Box plot 8](#_Toc181727868)

[2.3. Correlation Matrix Heatmap 9](#_Toc181727869)

[2.4. Scatter plot 9](#_Toc181727870)

[2.5. Interactive Map 10](#_Toc181727871)

[2.6. Density Heatmap 11](#_Toc181727872)

[3. Statistics 11](#_Toc181727873)

[3.1. Distribution 11](#_Toc181727874)

[3.1.1. Uniform Distribution 11](#_Toc181727875)

[3.1.2. Normal Distribution 12](#_Toc181727876)

[3.1.3. Poisson Distribution 12](#_Toc181727877)

[3.1.4. Kurtosis 13](#_Toc181727878)

[3.1.5. Skewness 13](#_Toc181727879)

[3.2. Central Tendency 13](#_Toc181727880)

[3.2.1. Mean 14](#_Toc181727881)

[3.2.2. Mode 14](#_Toc181727882)

[3.2.3. Median 14](#_Toc181727883)

[3.2.4. Min, Max and Range 15](#_Toc181727884)

[3.2.5. Standard Deviation 15](#_Toc181727885)

[3.2.6. The Five Number Summary 16](#_Toc181727886)

[3.3. Measures of Position 16](#_Toc181727887)

[3.3.1. Percentiles 16](#_Toc181727888)

[3.3.2. Interquartile Range (IQR) 17](#_Toc181727889)

[3.3.3. Z-Score 17](#_Toc181727890)

[4. Removing Outliers 18](#_Toc181727891)

[4.1. Imputation Method 18](#_Toc181727892)

[5. Machine Learning 18](#_Toc181727893)

[5.1. K-Means Clustering 19](#_Toc181727894)

[5.1.1. The Elbow Method 19](#_Toc181727895)

[5.1.2. The Silhouette Score 20](#_Toc181727896)

[5.2. Linear Regression 20](#_Toc181727897)

[5.3. Decision Tree Regressor 21](#_Toc181727898)

[5.4. Comparison of Regression Models 22](#_Toc181727899)

[5.5. Hyperparameter Tuning Using GridSearchCV 22](#_Toc181727900)

[6. Programming 23](#_Toc181727901)

[7. References 24](#_Toc181727902)

# Exploratory Data Analysis

For this project, I utilized three distinct datasets, each providing key insights into different aspects of tourism in Ireland. The first contains information on accommodations across the country, including hotels, B&Bs, and camping sites. The second dataset catalogues a variety of attractions in each Irish county. The third dataset details activities available throughout Ireland. Together, these datasets form a complete view of accommodations, attractions and activities, enabling deeper analysis of tourism patterns across Ireland.

These datasets were downloaded from <https://data.gov.ie/dataset/>

## Irish Accommodation Dataset

The first step is to import the pandas library, which allows us to read the CSV file and load it into a DataFrame called data.

By using the head() command, I displayed the first few rows of the loaded file, allowing us to verify that the data has been imported correctly.

Result:

A screenshot of a computer

Description automatically generated

Here, we can observe several interesting features, as well as some variables that we will not be using. I used the .drop() command to remove those unnecessary variables. Additionally, I renamed the remanding columns for an easier access throughout the coding process

After completing this step, I used the .info() command to visualize the data type of each feature ,and also, identify any null or missing observation.

I observed a few key details, there are some missing values, data types are floats and strings, and the unnecessary columns have been successfully dropped.

Next, I used .isnull().sum() to count all the missing values in each column.

I clearly observed that there are 184 missing values in the ‘town’ feature. I have decided not to remove any of those rows and will keep them all. Since my calculation will be per county in Ireland, missing the ‘town’ value is irrelevant.

I applied the .value\_counts() command to obtain the total count of accommodation in each county in Ireland. Additionally, I have created a new DataFrame called ‘acco’.

Finally, I plotted a line graph to visualize the total number accommodation per county.

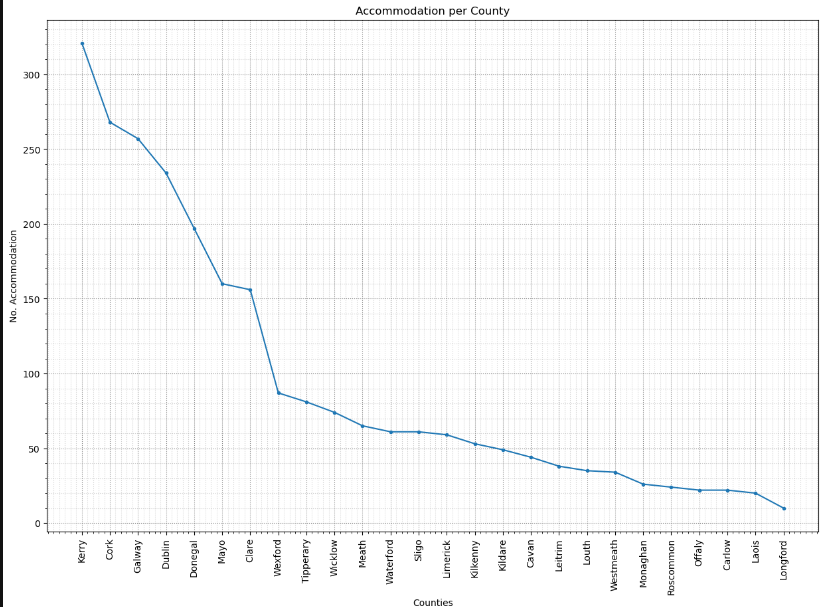


Figure 1. Graph of Accommodation per county in Ireland.

 For a detailed explanation of the process and code used for ‘accommodation dataset’, please refer to *Section 1.1. Irish Accommodation* in the accompanying Jupyter Notebook.

## Irish Attractions Dataset

Similar to the previous dataset I started importing pandas library in order to read the CSV file, I created a new DataFrame called attractions.

Continuously, I corroborated that the CSV file was loaded correctly by displaying the DataFrame(‘attractions’) using .head() command.

Result:

A screenshot of a computer

Description automatically generated

The next step involved removing columns that were unnecessary for the analysis of the data. Additionally, I renamed the remaining columns to ensure an easier access and improve readability throughout the coding process.

I checked for missing values using the .isnull().sum() command. Similar to the previous dataset, I observed some missing data in the ‘town’ feature. Noticed that, I decided not to erase any of those rows, in order to have more observations in my analysis.

I used the .value\_counts() command to group and count the total number of attractions per county.

Finally, to gain an effective visualization, I plotted the total number of attractions per county using the .plot(kind=’bar’) command.

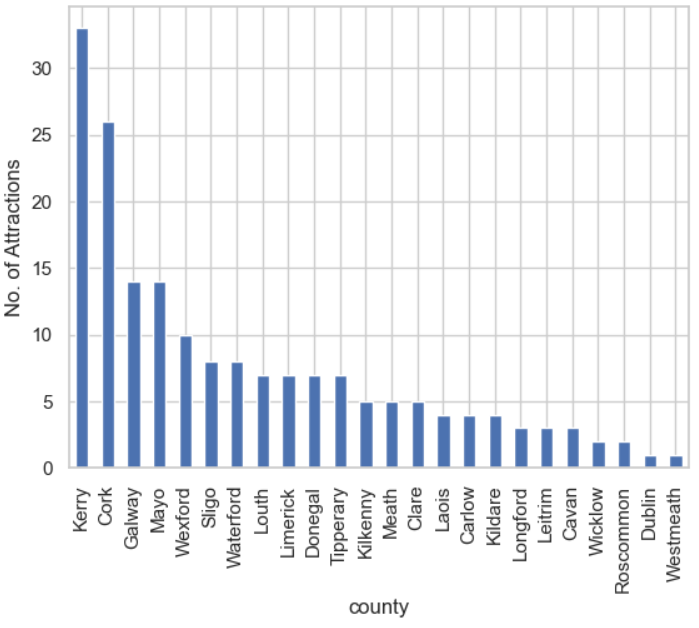


Figure 2. Graph of Attractions per county in Ireland.

 For a detailed explanation of the process and code used for ‘attractions dataset’, please refer to *Section 1.2. Irish Attractions* in the accompanying Jupyter Notebook.

## Irish Activities Dataset

I continued with exploratory data analysis (EDA) for the final dataset on activities in Ireland. I Imported the pandas library, read the csv. Commanded head.() to preview the loaded CSV file.

Result:

A screenshot of a computer

Description automatically generated

I erased the non-relative features, using .drop() code.

I commanded .info() to analyse information about the dataset. I observed that the features ‘Longitude’ and ‘Latitude’ are numerical values(float), while, ‘Name’, ‘AddressReggion’, and ‘AddressLocality’ are objects(string) values.

Furthermore, I renamed the remaining columns to an easier access and readability.

I used the .isnull() command to see whether there are any NaN values. Also, the command .sum() to count all the missing values.

Similar to our previous datasets, this one also has some missing values in the 'town' feature. However, I chose to keep all of these rows and did not remove any for my analysis.

Next step, I counted all the values in order to have the total activities per county, I used .value\_counts() command. As same as, I regrouped them into a new DataFrame called ‘act’.

For my last step, I plotted a bar graph by using .plot(kind=’bar’).

A graph of number of cities

Description automatically generated

Figure 3. Graph of Activities per county in Ireland.

 For a detailed explanation of the process and code used for ‘activities dataset’, please refer to *Section 1.3. Irish Activities* in the accompanying Jupyter Notebook.

## Creating new one Dataset

With the purpose of have all the data stored in the same place, I created a new DataFrame called combined\_df. I used the .concat() command to attach all the information. Additionally, I renamed each column with its respective title, such as ‘accommodation’, ‘activities, and ‘attractions’ for clarity reference.

For this combined dataset, I focused specifically on the total number of accommodations, activities and attraction per each county.

With all the information consolidated, I realized that there were two missing values in the ‘attractions’ feature. I lacked information about attractions in two counties, so I decided to use the .interpolate command to fill in these gaps and support my future analysis.

Finally, I plotted a scatter graph to gain a better understanding of the data.

A screen shot of a graph

Description automatically generated

Figure 4. Graph of Accommodation, Activities and Attractions per county in Ireland.

 For a detailed explanation of the process and code used for ‘combined\_df dataset’, please refer to *Section 1.4. Creating a new dataset* in the accompanying Jupyter Notebook.

### EDA Conclusion

In this process, I explored three different datasets by using **data cleaning** and **imputation** methods, to prepared them for analysis. After cleaning, I combined these three datasets into a single DataFrame by using **concatenation,** creating a new DataFrame ready for **machine learning** algorithms. This summarizes the key steps completed in the **Exploratory Data Analysis** (EDA) phase.

# Visualization

For this part of my project, I added the analysis with a variety of plots and heatmaps to explore and identify potential relationships between the features on the dataset. Additionally, I incorporated interactive maps to provide an easier way to visualize all the accommodation, activities and attractions available in Ireland. These visualizations help in spotting trends, correlations, and distribution patterns across the different features.

## Pair plot

I decided to begin with a pair plot (scatterplot matrix), to generate an initial understanding of the relationships within the dataset. I simply import seaborn library and used .pairplot() command to create this matrix of graphs. The pair plot provides a combination of histogram and scatter plots, enabling an overview of the dataset’s distribution patterns and correlation.

A group of blue and white graphs

Description automatically generated

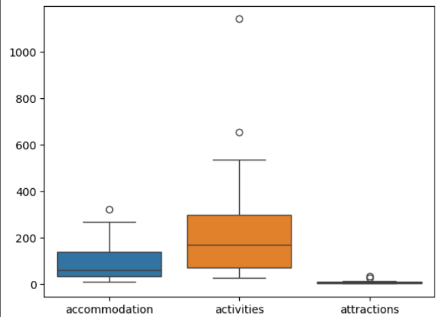
Figure 5. Pair plot within Accommodation, Activities and Attractions in Ireland.

 For a detailed explanation of the process and code used for ‘Pair plot’, please refer to *Section 2.1. Pairplot* in the accompanying Jupyter Notebook.

## Box plot

Another useful graph that I employed is the box plot, which I created commanding .boxplot(). This type of graph is particularly useful to identifying outliers. It effectively visualizes the possible outliers on the dataset.

A graph of different colored squares

Description automatically generated  
Figure 6.1. Graph of Total values Figure 6.2. Graph of Mean Values

For a detailed explanation of the process and code used for ‘Box plot’, please refer to *Section 2.2. Box plot* in the accompanying Jupyter Notebook

## Correlation Matrix Heatmap

With a heatmap I observed a strong correlation between ‘accommodation’ and ‘activities’, indicating that areas with more available activities tend to have more accommodation. Additionally, there is a moderate correlation with ‘attractions’, suggesting that also popular tourist destination increment the accommodation development.

A screenshot of a computer screen

Description automatically generated

Figure 7. Heatmap of Accommodation, Activities and Attractions in Ireland.

For a detailed explanation of the process and code used for ‘Correlation Matrix Heatmap’, please refer to *Section 2.3.Correlation Matrix Heatmap* in the accompanying Jupyter Notebook

## Scatter plot

I have created a Scatter plot to visualize the location of Accommodation, Attractions and Activities across Ireland. For this section I used the ‘longitude’ and ‘latitude’ columns for pinpoint exact locations.

This scatter plot effectively illustrates the distributions of those features across the country, revealing notable patterns. For instance, there is a high concentration of activities in County Dublin, which likely reflects a popularity of the area as a tourist destination.

A map of different colored dots

Description automatically generated

Figure 8.Scatter Plot of Accommodation, Activities and Attractions in Ireland.

For a detailed explanation of the process and code used for ‘Scatter plot’, please refer to *Section 2.4. Scatter plot* in the accompanying Jupyter Notebook

## Interactive Map

I added an interactive map, which provides users with a convenient tool to explore every county and town in Ireland. With this map, users can easily view all available destinations, including accommodation, activities, and attractions. This interactive feature offers and effective and user-friendly way to explore the Emerald Isle.

For this report, I attached three images of the map, the first one showing ‘Accommodations’ in green colour, the second one showing ‘Attractions’ in red colour, the third one showing ‘Activities’ in blue colour.

|  |
| --- |
| Accommodation Attractions Activities |

A map of england with green circles

Description automatically generated  A map of the united kingdom

Description automatically generated

Figure 9.1. Figure 9.2. Figure 9.3.

For a detailed explanation of the process and code used for ‘Interactive Map’, please refer to *Section 2.5. Interactive map* in the accompanying Jupyter Notebook

## Density Heatmap

Furthermore, I developed a heatmap to visualize the areas with the highest concentrations of accommodations, activities and attractions. This heatmap highlights regions with dense clusters of tourist offerings, providing a clear view of popular areas and helping to identify potential hotspots for tourism across the island.

A map of the world

Description automatically generated

Figure 10. Density Heatmap of Accommodation, Activities and Attractions in Ireland.

For a detailed explanation of the process and code used for ‘Density Heatmap’, please refer to *Section 2.6. Density Heatmap* in the accompanying Jupyter Notebook

# Statistics

Statistics play a fundamental role in the data analysis field, providing essential tools and techniques for understanding and preparing the data. By analysing measures of **central tendency**, **dispersion**, **shape**, I gain insight into the data’s **distribution** and variability. These measures help identify patterns, **detect outliers**, and guide data preprocessing, such as **imputation** and **normalization**. Trough these statistical analyses, I can better prepare the data for model training, optimize feature selection and enhance model interpretability and accuracy.

## Distribution

### Uniform Distribution

I began by plotting graphs to evaluate whether any of the datasets (accommodations, activities, attractions) followed a **uniform distribution**. Every histogram shows the distribution of the total value per county, with an added **Kerner Density Estimate (KDE**). Although it appears relatively spread across the range, higher and lower counts, peaks and dips, this indicate that the distribution is **not uniform**.

A graph of a number of blue bars

Description automatically generated with medium confidenceA graph of activities with numbers and lines

Description automatically generated with medium confidenceA graph of a number of people

Description automatically generated

Figure 11.1 Uniform distribution of Accommodation. Figure 11.2. Uniform distribution of Activities. Figure 11.3. Uniform distribution of Attractions.

For a detailed explanation of the process and code used for ‘Uniform Distribution’, please refer to *Section 3.1.1 Uniform Distribution* in the accompanying Jupyter Notebook

### Normal Distribution

Consequently, I plotted histograms to examine whether the data could fit a **normal distribution**, the **KDE line** has been added to help clarify density variations, highlighting regions with peaks and dips.

However, the data **does not follow a normal distribution**. Instead, it displays a right-skewed distribution in all of them.

A graph of a normal distribution

Description automatically generated A graph of activities

Description automatically generated A graph with blue lines and a blue line

Description automatically generated

Figure 12.1. Normal distribution of Accommodation. Figure 12.2. Normal distribution of Activities. Figure 12.3. Normal distribution of Attractions.

For a detailed explanation of the process and code used for ‘Normal Distribution’, please refer to *Section 3.1.2 Normal Distribution* in the accompanying Jupyter Notebook

### Poisson Distribution

These histograms appear to illustrate an attempt to model **a Poisson distribution** for the dataset (accommodation, attraction, activities), with the parameter λ (lambda) representing the average group size. The probability of each count is calculated on an average rate of occurrence (λ).

The **KDE line** provides a smooth curve that emphasises the central tendency and spread of the data. This visualisation suggests that most counts fall close to **the mean**.

A graph of a number of accommodation

Description automatically generated A graph of a number of activities

Description automatically generated A graph of a number of attraction

Description automatically generated

Figure13.1. Poisson distribution of Accommodation. Figure 13.2. Poisson distribution of Activities. Figure 13.3. Poisson distribution of Attractions.

For a detailed explanation of the process and code used for ‘Poisson Distribution’, please refer to *Section 3.1.3 Poisson Distribution* in the accompanying Jupyter Notebook

### Kurtosis

Using **Kurtosis** to analyse the distribution of the datasets (accommodations, activities, and attractions) can give insights of the tail behaviour of the data.

These are the results:

* Kurtosis of Accommodation: 0.22946016701505556
* Kurtosis of Activities: 5.02537121724218
* Kurtosis of Attractions: 4.671342211366463

A kurtosis greater than 3 will indicate a **Positive Kurtosis**, which suggest a heavy-tailed distribution with potential outliers.This is the case of the ‘activities’ and ‘attractions’ datasets, both have **kurtosis** values exceeding 3.

In the case of ‘accommodation’, the **Kurtosis** is close to 0, indicating a normal distribution.

The next histogram effectively highlights the imbalance in the data.

A graph of a distribution

Description automatically generated

Figure 14. Distribution of Accommodation, Activities and Attractions according to Kurtosis.

For a detailed explanation of the process and code used for ‘Kurtosis’, please refer to *Section 3.1.4 Kurtosis* in the accompanying Jupyter Notebook

### Skewness

In addition, I computed Skewness to gain a second perspective on the distribution patterns. The results are:

Accommodation dataset with a **Skewness** of **1.24**, this data shows a moderate positive skew’.

Activities dataset, the **skewness** values is **2.18** indicating a strong positive skew, meaning that the distribution is heavily right-skewed.

Attraction dataset, similarly, the **Skewness** of **2.15** also indicates a strong positive skew.

For a detailed explanation of the process and code used for ‘Skewness’, please refer to *Section 3.1.5 Skewness* in the accompanying Jupyter Notebook.

## Central Tendency

To gain an understanding of the typical values, I analysed measures of central tendency as well as Mean, Mode, Median, Min, Max, Range, and Standard Deviation. These measures represent the centre point of the data distribution.

### Mean

I calculated the mean of each dataset by creating a custom function according to the formula.

A mathematical equation with a number and a number

Description automatically generated with medium confidence

These are the results:

* Mean of accommodation in Ireland: 94.53846153846153
* Mean of activities in Ireland: 234.8846153846154
* Mean of attractions in Ireland: 7.625

For a detailed explanation of the process and code used for ‘Mean’, please refer to *Section 3.2.1Mean* in the accompanying Jupyter Notebook

### Mode

Similarly, I created a function in order to calculate the **Mode** of the datasets.

A math equation with numbers and symbols

Description automatically generated

The results are:

* Mode of accommodation in Ireland: Two modes found: [61, 22]
* Mode of activities in Ireland: No unique mode
* Mode of attractions in Ireland: [7]

For a detailed explanation of the process and code used for ‘Mode’, please refer to *Section 3.2.2Mode* in the accompanying Jupyter Notebook

### Median

Additionally, I coded the **Median** Formula.

A math equations with black text

Description automatically generated with medium confidence

Results:

* Median of accommodation in Ireland: 60.0
* Median of activities in Ireland: 168.5
* Median of attractions in Ireland: 5.0

For a detailed explanation of the process and code used for ‘Median’, please refer to *Section 3.2.3 Median* in the accompanying Jupyter Notebook

### Min, Max and Range

Furthermore, I calculated the **Min, Max and Range** for every respective feature.

Results:

Min:

* accommodation 10.0
* activities 27.0
* attractions 1.0

Max:

* accommodation 321.0
* activities 1140.0
* attractions 33.0

Range:

* accommodation 311.0
* activities 1113.0
* attractions 32.0

I observed a significant **range** value in the ‘Activities’ and ‘Accommodation’ features, suggesting that outliers may be present.

For a detailed explanation of the process and code used for ‘Min, Max, and Range’, please refer to *Section 3.2.4 Min, Max and Range* in the accompanying Jupyter Notebook

### Standard Deviation

For the calculation of Standard Deviation, I decided to create a function called calc\_stdev. This is the code according to the formula.

A square root of a mathematical equation

Description automatically generated

These are the results:

* Standard Deviation of accommodation in Ireland: 87.81196290029261
* Standard Deviation of activities in Ireland: 242.81859752040236
* Standard Deviation of attractions in Ireland: 7.487837360235153

The standard deviation of Accommodation and for Attractions reflects a significant **spread** in the data.

For a detailed explanation of the process and code used for ‘Standard Deviation’, please refer to *Section 3.2.5 Standard Deviation* in the accompanying Jupyter Notebook

### The Five Number Summary

To conclude the analysis of **Central Tendency**, I commanded **.describe()**, which provides key summary statistics as the count, mean, standard deviation, min, max and percentiles.

A table with numbers and text

Description automatically generated

This computation offers an additional perspective, enabling a comprehensive view of the data’s spread and central values.

For a detailed explanation of the process and code used for ‘The Five number summary’, please refer to *Section 3.2.6 The Five Number Summary(df.describe()) in the* accompanying Jupyter Notebook

## Measures of Position

For further insights, I examined measures of position such as percentiles, quartiles, and Z-score. These measures provide information on the relative standing of data points within each feature, helping us understand the data distribution.

### Percentiles

I examined percentiles within each feature. Percentiles dive the data into 100 equal parts, showing the position of a particular point relative to the rest.

Results:

Accommodation feature:

* 25th Percentile of Accommodation: 34.25
* 75th Percentile of Accommodation: 138.75

Attraction feature:

* 25th Percentile of Attractions: 3.0
* 75th Percentile of Attractions: 8.0

Activities feature:

* 25th Percentile of Activities: 70.75
* 75th Percentile of Activities: 296.75

For a detailed explanation of the process and code used for ‘Percentiles’, please refer to *Section 3.3.1 Percentiles and IQR* in the accompanying Jupyter Notebook

### Interquartile Range (IQR)

Additionally, in order to identify potential outliers, I calculated the Interquartile Range (IQR). The IQR is the difference between the 75th percentile and the 25th percentile, measuring the spread of the middle of the 50% of the data.

Results:

Accommodation feature:

* IQR: 104.5

Attraction feature:

* IQR: 5.0

Activities feature:

* IQR: 226.0

These IQR values reflect a notable broader spread in the ‘activities and ‘accommodation’ features, suggesting higher variability within them. ‘Attraction’ feature has a narrow IQR, indicating a more consistent distribution of values.

For a detailed explanation of the process and code used for ‘Interquartile Range (IQR)’, please refer to *Section 3.3.1 Percentiles and IQR* in the accompanying Jupyter Notebook

#### Outliers

IQR range is particularly useful for detecting variability by extreme values or outliers. Data points falling below Q1 – 1.5 \* IQR of above Q3 + 1.5\*IQR are considered potential outliers.

Results

Accommodation feature:

* Kerry 321

Activities feature:

* Dublin 1140
* Cork 653

Attractions feature:

* Kerry 33
* Cork 26

Those are possible outliers according to the IQR detection of outlier formula.

For a detailed explanation of the process and code used for ‘Outliers’, please refer to *Section 3.3.2.1 Outliers Using IQR* in the accompanying Jupyter Notebook

### Z-Score

Another way to detect potential outliers and understand how individual data points compare to the dataset as a whole, I calculated Z-score for each feature. The Z-score measures the number of **standard deviations** a data point is from the mean. Typically, data points with a Z-score above 3 or below -3 are considered outliers.

Results:

Accommodation:

* Not found

Attractions:

* Kerry 33

Activities:

* Dublin 1140

These are the possible outliers according to **Z-score**

For a detailed explanation of the process and code used for ‘Z-Score’, please refer to *Section 3.3.3 Z-Score* in the accompanying Jupyter Notebook

# Removing Outliers

To improve the reliability of the analysis and model performance, I created a new DataFrame called c\_no\_outliers, which excludes identified outliers from the last step. This DataFrames will allow more stable results in certain **Machine Learning Algorithms** that are sensitive to extreme values**.**

I applied the **IQR** method to remove the outliers. I chose this approach because the **Z-Score** method is particularly useful for detecting outliers when data follow a normal distribution. Conversely, The **IQR** method is more effectively for non-normally distributed data.

 For a detailed explanation of the process and code used to remove outliers, please refer to *Section 4. Removing Outliers* in the accompanying Jupyter Notebook.

## Imputation Method

After removing outliers, I addressed missing data by applying an **Imputation Method.** Using the fillna() functions in combination with .mean(), I Replaced missing values with the mean of each respective feature into a new dataset called ‘df\_no\_outliers.’

For a detailed explanation of the process and code used to remove outliers, please refer to *Section 4.1 Imputation Method* in the accompanying Jupyter Notebook.

# Machine Learning

In this project, The **CRISP-DM** framework is highly suitable, it provides a comprehensive and flexible methodology that are need in data analysis. CRISP-DM consist of Business understanding, Data understanding, Data preparation, Modelling evaluation, and Deployment.

**Supervised Learning** is suitable for predicting accommodation levels based on other variables like activities and attractions. The project has a clear target variable (accommodation) and label data (number of activities, attractions) that can be used as predictors.

**Machine learning techniques** are applied to predict accommodation availability in each area. I experimented with three models: K-Means Clustering, Linear Regression, and Decision Tree Regressor. Each model offers a unique approach to analysing the relationship between accommodation, attractions and activities, with clustering providing insights into group patterns and the regressor aiming to predict accommodation numbers. To ensure optimal model performance, I also implemented GrindSearchCV for hyperparameter turning across multiple regression models, allowing for a comprehensive comparison of model accuracy and helping identify the best-performing approach. This comparative analysis not only enhances prediction accuracy but also provides a deeper understanding of which features and models capture the best relationship within the data.

## K-Means Clustering

In this study, I applied **K-means clustering** to ‘df\_no\_outliers’ dataset to uncover patterns in accommodation, activities, and attractions. I selected this dataset(df\_no\_outliers) due to **K-means** is a method sensitive to outliers or noise.

Using clustering, I aimed to group locations with similar tourism characteristics, revealing distinct types of tourism regions. This analysis highlights two primary clusters: one with high accommodation and activity counts, and another with lower counts, indicating less touristic areas.

A graph of different colored dots

Description automatically generated

Figure 15. Scatter plot of clusters with its centroids.

**Interpretation:**

**Class 1(Gray):** This has lower counts of both activities and accommodations. It could represent rural regions.

**Class 2(Blue):** This group has higher counts of both activities and accommodations. Many points are concentrated in the higher ranges of both axes. This cluster represents tourism-heavy or urban areas with significant attractions and a developed infrastructure.

For a detailed explanation of the process and code used to K-Means Clustering, please refer to *Section 5.1 K-means* in the accompanying Jupyter Notebook.

### The Elbow Method

In other to determine the most appropriate number of clusters, I used two validation techniques: the elbow method and the silhouette score. The elbow method involves plotting the inertia (sum of squared distances between data points and their cluster centroids) for different values of ***k***. The graph displayed a clear ‘elbow’ at k = 2.

A graph with a line

Description automatically generated

Figure 16. Line graph of number of clusters.

For a detailed explanation of the process and code used to ‘The Elbow Method’, please refer to *Section 5.1.1 Elbow method and silhouette score* in the accompanying Jupyter Notebook.

### The Silhouette Score

To further validate this choice, I calculated the **silhouette score**, which measures how well each data point is grouped within its assigned cluster. A score closer to 1 indicates well-defined, while a score closer to -1 suggest poor separation. With a **silhouette score of 0.623**, I confirmed thar two clusters offer a strong grouping with good separation.

For a detailed explanation of the process and code used to ‘The Silhouette Score’, please refer to *Section 5.1.1 Elbow method and silhouette score* in the accompanying Jupyter Notebook.

## Linear Regression

In this project, I used the ‘combined\_df’ dataset to predict **accommodation values** based on the features **activities** and **attractions**. To achieve this prediction, I applied **Linear Regression** model. Linear regression is a technique for predictive analysis.

I chose this dataset because it represents the original data on tourism metrics across different counties and provides a comprehensive view of tourism throughout Ireland. In addition, and **linear regression** is a method robust to **outliers**.

**Model Setup**

* **Features (X):** I used ‘activities’ and ‘attractions’ as the input features.
* **Target (y):** The target variable is ‘accommodation’ representing the number of accommodations per county.

I trained a model that satisfactory predicts the **accommodation count** per county based on the number of activities and attractions available.

A graph of activity and activity

Description automatically generated

Figure 17. Scatter Plot of Attractions and Activities according to Accommodation.

**Model Evaluation**

To evaluate the performance of the model and ensure that it makes accurate predictions, I used the R2 Score as metric. The **R2 Score** measures the proportion of the variance in the target variable that is predictable form the input features. An R2 score closer to 1 indicates a better fit.

* The model archived an **R2 score of 90.68%**, indicating that the model is operating effectively and provides reliable predictions.

For a detailed explanation of the process and code used to ‘Linear Regression’, please refer to *Section 5.2. Linear Regression* in the accompanying Jupyter Notebook.

## Decision Tree Regressor

In addition, I applied **Decision Tree Regressor** in order to predict **accommodation counts**, based on the activities and attractions available on each county.

**Model Setup**

* **Features (X):** ‘activities’ and ‘attractions’ as the input features.
* **Target (y):** The target variable is ‘accommodation’ representing the number of accommodations per county.

I trained a model that **predicts** the accommodations per county based on activities and attractions.

A diagram of a structure

Description automatically generated

Figure18. Decision tree predicting Accommodation.

**Model Evaluation**

Similarly to the previous model, I used the R2 Score as metric.

* The model archived an **R2 score of 100%**, indicating that the model is operating effectively and provides reliable predictions.

For a detailed explanation of the process and code used to ‘Decision Tree Regressor’, please refer to *Section 5.3. Decision Tree Regressor* in the accompanying Jupyter Notebook.

## Comparison of Regression Models

I explored multiple regression models to predict the number of accommodations available in each country. By comparing various models, I aimed to identify the best approach for accurately predicting accommodations.

The models evaluated included **Linear Regression, Decision Tree Regressor, Random Forest Regressor, K-Nearest Neighbors, Support Vector Machine**, and **Random Forest Classifier** as an experimental comparison, though it is not designed for continuous target prediction.

Results:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **MAE** | **MSE** | **RMSE** | **R2 Score** |
| Linear Regression | 16.27 | 336.70 | 18.35 | 0.967 |
| Decision Tree | 21.67 | 763.00 | 27.62 | 0.925 |
| Random Forest | 23.22 | 989.73 | 27.62 | 0.903 |
| K-Nearest Forest | 38.07 | 4054.06 | 63.67 | 0.601 |
| Support Vector Machine | 58.18 | 11545.30 | 107.45 | -0.137 |
| Random Forest Classifier | 25.00 | 1090.00 | 33.02 | 0.893 |

Table 1.

**Conclusion:**

**Best model:** Linear Regression with the lower error metrics and the highest R2 score.

**Decision Tree Regressor**: It is a good alternative, although it has a higher error metrics that linear regression.

**Poor Performers**: k-NN and SVM both perform poor performance in this context.

**Random Forest Classifier**: it is the incorrect model to use, as classifier is designed for categorical outputs.

For a detailed explanation of the process and code used to ‘Comparison of Regression Models’, please refer to *Section 5.4. Comparison of Regression Models* in the accompanying Jupyter Notebook.

## Hyperparameter Tuning Using GridSearchCV

To improve the performance of each regression model, I used Grid Seach with cross-validation to find the optimal hyperparameters.

The models evaluated are **Lasso Regression, Decision Tree Regressor, Random Forest Regressor, k-Nearest Neighbors (KNN), Support Vector Regressor (SVR).**

|  |  |  |
| --- | --- | --- |
| **Model** | **Best Parameters** | **Best CV R2 Score** |
| Lasso | alpha:10.0 | 0.248372 |
| Decision Tree | max\_depth: 10 | 0.843295 |
| Random Forest | N\_estimators:200 | 0.838110 |
| K-Nearest Neighbors (KNN) | Metric: minkowski, n\_neighbors:3, p:2 | 0.854531 |
| Support Vector Regressor (SVR) | C:100, gamma: 0.01, kernel: rbf | -0.773829 |

**Table 2.**

**Conclusion:**

The hyperparameter tuning process revealed that **K-Nearest Neighbors (KNN)** has the highest cross-validated R2 score, followed by the **Decision tree** and **Random Fores** models.

On the other hand, **Lasso Regression** and **SVR** did not perform well.

For a detailed explanation of the process and code used to ‘Hyperparameter Tuning using GridSearchCV, please refer to *Section 5.5. Hyperparameter Tuning using GridSearchCV* in the accompanying Jupyter Notebook.

# Programming

Programming is the core of this project, enabling data handling, analysis, and the implementation of machine learning models. Throughout my assignment, I used various programming paradigms, each offering unique advantages and influencing design choices and problem-solving strategies.

**Procedural programming** played a significant role in the data processing phase. Using the procedural capabilities of Python, I created sequences of steps to clean, transform, and merge datasets. For example, I applied functions like fillna() and drop() to handle missing values and irrelevant columns, and also, clear and make the code simple and readable.

**Object-oriented programming** was beneficial for organizing code into classes when I was working with machine learning models. By defining classes for models and evaluation metrics, I was able to encapsulate functionality, making the codebase more organized and reusable. For instance, creating a class model evaluation that includes methods for computing metrics like **Silhouette Score, Mean Squared Error,** and plotting results helped to made testing multiple models, as **K-means** and **Linear Regression**, more efficient.

Functional programming techniques also helped in streamlining data transformations. Python’s lambda functions and map() allowed for concise data processing, such as applying transformations across columns without needing to define new variables. Additionally, using functional approaches like apply() with lambda expressions simplified complex data filtering and feature engineering task. This helped make code more expressive and less error-prone.

These paradigms shaped my design decision across the project. The procedural approach kept data preprocessing manageable, object-oriented programming added structure for model handling and evaluation, and functional programming enhanced code conciseness. These combinations of paradigms improved both the efficiency and readability of the code, contributing to a solution for predicting accommodations based on attractions and activities.

# References

**Books and Publications**

* Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J., 1984. *Classification and Regression Trees*. Wadsworth, Belmont, CA.
* Chen, G. and Astebro, T., 2003. How to deal with missing categorical data: Test of a simple Bayesian method. *Organizational Research Methods*, 6(3), pp.309-327.
* Chen, C.H., and Others, 2016. *Handbook of Data Visualization*. Berlin: Springer.
* Dangeti, P., Yu, A., Chung, C., and Yim, A., 2018. *Numerical Computing with Python*. Packet Publishing.
* Géron, A., 2019. *Hands-on Machine Learning with Scikit-Learn and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems*. 2nd ed. O'Reilly Media, Inc.
* Healy, K., *Data Visualization: A Practical Introduction*. Princeton, NJ: Princeton University Press.
* Kirk, M., 2014. *Thoughtful Machine Learning*. O'Reilly Media, Inc.
* Larose, D., and Larose, C., 2014. *Discovering Knowledge in Data: An Introduction to Data Exploration*. 2nd ed. John Wiley and Sons, Inc.
* McKinney, W., 2012. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. Sebastopol: O’Reilly Media.
* Müller, A.C. and Guido, S., 2017. *Introduction to Machine Learning with Python*. O'Reilly Publishers.
* Pyle, D., 1999. *Data Preparation for Data Mining*. Morgan Kaufmann, San Mateo, CA.
* Swamynathan, M., 2019. *Mastering Machine Learning with Python in Six Steps: A Practical Implementation Guide to Predictive Data Analytics Using Python*. Apress.
* Tan, P.N., Steinbach, M., Karpatne, A., and Kumar, V., 2019. *Introduction to Data Mining*. 2nd ed. Pearson.
* Theobald, O., 2024. *Machine Learning with Python*. Packt Publishing.
* Zaki, M.J., and Meira, W., 2020. *Data Mining and Machine Learning: Fundamental Concepts and Algorithms*. Cambridge, CB2 8BS, United Kingdom.

**Online Articles**

* Contreras, J.A.L., 2021. *Data Analysis and Linear Regression*. Analytics Vidhya. Available at: <https://medium.com/analytics-vidhya/data-analysis-and-linear-regression-567da388c2a1> [Accessed 1 April 2024].
* Grant, P., 2021. *Model Validation and Testing: A Step-by-Step Guide*. [online] Built In. Available at: <https://builtin.com/data-science/model-validation-test>.
* Halilovic, I., 2021. *Markdown for Jupyter Notebooks CheatSheet*. Medium. Available at: <https://medium.com/@inge.halilovic/markdown-for-jupyter-notebooks-cheatsheet-386c05aeebed> .
* Kjytay, J., 2021. *The Box-Cox and Yeo-Johnson Transformations for Continuous Variables*. [online] Statistical Odds & Ends. Available at: <https://statisticaloddsandends.wordpress.com/2021/02/19/the-box-cox-and-yeo-johnson-transformations-for-continuous-variables/>.
* Lakshanagv, L., 2021. *Cross-Validation Techniques in Machine Learning for Better Model Performance*. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2021/05/4-ways-to-evaluate-your-machine-learning-model-cross-validation-techniques-with-python-code/>.
* Lee, E., 2019. *An Intro to Hyper-parameter Optimization Using Grid Search and Random Search*. [online] Medium. Available at: <https://medium.com/@cjl2fv/an-intro-to-hyper-parameter-optimization-using-grid-search-and-random-search-d73b9834ca0a>