**MSc in Data Analytics**

**CA2**

Author: Maria Koleva

E-mail: sba23020@student.cct.ie

Student ID: 23020

GitHub: https://github.com/mkoleva0/DataAnalytics\_CA2

Table of Contents

[List of Figures 3](#_Toc155554841)

[Abstract 4](#_Toc155554842)

[Introduction 4](#_Toc155554843)

[Project Planning 4](#_Toc155554844)

[Agile Framework Adoption 4](#_Toc155554845)

[Tools and Technologies 5](#_Toc155554846)

[Project Timeline 5](#_Toc155554847)

[Data Acquisition and Licensing 5](#_Toc155554848)

[Dublin Dataset Description 5](#_Toc155554849)

[Dublin Dataset Licensing and Permissions 5](#_Toc155554850)

[Madrid Dataset Description 5](#_Toc155554851)

[Madrid Dataset Licensing and Permissions 6](#_Toc155554852)

[Comments Dataset Description 6](#_Toc155554853)

[Comments Dataset Licensing and Permissions 6](#_Toc155554854)

[Data Loading and Initial Assessment 6](#_Toc155554855)

[Dublin Dataset 6](#_Toc155554856)

[Madrid Dataset 7](#_Toc155554857)

[Exploratory Data Analysis 9](#_Toc155554858)

[Dublin hourly volatility 9](#_Toc155554859)

[Strategies for improvement 10](#_Toc155554860)

[Madrid hourly volatility 10](#_Toc155554861)

[Similarities between Dublin and Madrid 10](#_Toc155554862)

[Inferential Statistics 11](#_Toc155554863)

[Data preparation and Normality Check 11](#_Toc155554864)

[Sampling 11](#_Toc155554865)

[Confidence Intervals 11](#_Toc155554866)

[Hypothesis Testing 11](#_Toc155554867)

[Findings 12](#_Toc155554868)

[Machine Learning Analysis 12](#_Toc155554869)

[Data importation and preliminary setup 12](#_Toc155554870)

[Data Preparation and Feature Development 12](#_Toc155554871)

[Model Selection and Refinement 12](#_Toc155554872)

[Linear Models with Feature Engineering 13](#_Toc155554873)

[Non-Linear Model – RandomForestRegressor 13](#_Toc155554874)

[Evaluation 13](#_Toc155554875)

[Decision Tree and Random Forest 14](#_Toc155554876)

[Sentiment Analysis 14](#_Toc155554877)

[Interactive Dashboard 15](#_Toc155554878)

[Conclusion and Recommendations 16](#_Toc155554879)

[Appendix A: Outlier Analysis 17](#_Toc155554880)

[A1: IQR Visualization for Dublin Dataset 17](#_Toc155554881)

[A2: IQR Visualization for Madrid Dataset 17](#_Toc155554882)

[Appendix B: Statistical Analysis Details 18](#_Toc155554883)

[B1: Data Alignment and pre-processing 18](#_Toc155554884)

[B2: Sampling Techniques 18](#_Toc155554885)

[Appendix C: Machine Learning Analysis Details 20](#_Toc155554886)

[C1: Code for pre-processing and Model Evaluation 20](#_Toc155554887)

[C2: Feature Engineering Techniques 20](#_Toc155554888)

[C3: Impact on Model Performance 20](#_Toc155554889)

[C4: Visualization of Model Performance 21](#_Toc155554890)

[C5: Reddit API call details 22](#_Toc155554891)

[C6: Word cloud visualization for sentiment analysis 23](#_Toc155554892)

[Appendix D: Data Loading 24](#_Toc155554893)

[D1: Dublin Dataset 24](#_Toc155554894)

[D2: Madrid Dataset 24](#_Toc155554895)

[References 26](#_Toc155554896)

# List of Figures

[Figure 1: First five rows of Dublin bike rental data for August 2022 7](#_Toc155554899)

[Figure 2: Pandas 'info()' output for Dublin bike rental data 7](#_Toc155554900)

[Figure 3: First five rows of Madrid bike rental data for August 2022 8](#_Toc155554901)

[Figure 4: Pandas ‘info()’ output for Madrid bike rental data 8](#_Toc155554902)

[Figure 5: Hourly volatility for the top five stations in Dublin 9](#_Toc155554903)

[Figure 6: Hourly volatility for the top five stations in Madrid 10](#_Toc155554904)

[Figure 7: Evaluation of the models 14](#_Toc155554905)

[Figure 8: Sentiment distribution of comments 15](#_Toc155554906)

[Figure 9: IQR analysis for Dublin 17](#_Toc155554907)

[Figure 10: IQR analysis for Madrid 17](#_Toc155554908)

[Figure 11: ECDF plot 19](#_Toc155554909)

[Figure 12: Performance of Random Forest Model with Day and Hour Features 21](#_Toc155554910)

[Figure 13: Performance of Linear Regression Model with OneHot Encoded Features 21](#_Toc155554911)

[Figure 14: Performance of Ridge model with Polynomial features 22](#_Toc155554912)

[Figure 15: Word cloud of sentiments from Dublin Bikes Reddit Comments 23](#_Toc155554913)

# Abstract

The project looks into rental bike systems in Dublin and Madrid. These two European capitals were chosen for comparison to see how bike-sharing works in different city environments. The main goal is to examine the usage patterns of rental bikes. The study begins with an analysis of bike rental data from Dublin and Madrid, identifying when and how often people rent bikes to understand unique and shared patterns across the cities. It uncovers that the busiest stations are located in the city centres, indicating a shared trend in bike rental behaviour between Dublin and Madrid.

An in-depth analysis of Dublin was performed to forecast bike availability using a range of regression-based machine learning models, and classification methods to categorize bike usage patterns. Additionally, an analysis of Reddit comments was conducted to gain insight into public opinion and sentiment about Dublin’s bike sharing system.

Word count: 143

# Introduction

Nowadays, bike-sharing has become a popular solution to improve transportation and reduce environmental impact. This study compares the bike-sharing systems in Dublin and Madrid to understand how they operate in different urban environments and how people use them.

Dublin, with its commitment to green living, has seen a significant rise in cycling, attributed to its supportive infrastructure and initiatives. The city's strategies in promoting cycling and improving related infrastructure are well-documented in its development plans (Dublin City Council, 2022, ch.8). On the other hand Madrid has recently initiated efforts to promote cycling among its residents. A key part of the plan is expanding BiciMAD, Madrid's bike-sharing service, by adding more stations across the city. This shift towards more sustainable urban mobility is outlined in the city's recent - Recovery, Transformation and Resilience Plan (City of Madrid, 2021).

This study aims to understand the patterns surrounding bike-sharing in these cities, offering insights into their operational successes and areas for improvement.

Word count: 159

# Project Planning

## Agile Framework Adoption

Adopting the Agile Project Management framework allows flexibility and adaptability in handling the complex nature of the cycling data analysis. The Agile Project Management framework is particularly well-suited to project that benefit from fast-paced, incremental progress, such as the dynamic analysis of the bike-sharing scheme data. By breaking down the project into smaller, more manageable segments, or “sprints”.

Word count: 58

## Tools and Technologies

The project utilizes Jupyter Notebook for an interactive data analysis and documentation environment, Python for its powerful computational capabilities and extensive libraries, and GitHub for version control and project tracking. These tools support an iterative, flexible approach to data analysis, aligning with the Agile framework's principles.

Word count: 46

## Project Timeline

The project is set for six-week duration, structured in sprints:

Week 1: Data acquisition

Week 2: Preliminary analysis to understand basic patterns

Week 3-4: In-depth analysis of Dublin bike rental scheme, using statistical models and machine learning techniques to predict usage patterns and assess system performance

Week 5: Comparative analysis with Madrid bike rental

Week 6: Final review and report submission

Word count: 61

## Data Acquisition and Licensing

### Dublin Dataset Description

The study uses the 'Dublinbike bike sharing scheme. Historical data for bike locations, August 2022' dataset (Dublin City Council, 2022). It includes data on rental times, station locations, and other relevant transport details.

Word count: 33

### Dublin Dataset Licensing and Permissions

The dataset is licensed under Creative Commons Attribution 4.0 (CC BY 4.0) by Dublin City Council (Creative Commons, 2021). This license allows for adaptation and distribution, provided that appropriate credit is given. The project will adhere to these terms by crediting Dublin City Council in any publications or outputs that use this data.

Word count: 53

### Madrid Dataset Description

The report uses the 'EMT Madrid Open Data' (EMT Madrid, 2022) for insights into Madrid's bike rental services. The dataset includes information on rental locations and patterns throughout August 2022.

Word count: 30

### Madrid Dataset Licensing and Permissions

The Madrid dataset is available under terms specified by EMT Madrid's Open Data portal. The licensing terms, provided in Spanish, have been summarized in English for this report. They state that the data can be used, shared, and adapted for any purpose, with the condition that EMT Madrid is acknowledged as the source. Full licensing terms can be accessed at EMT Madrid (2022), and the original terms are available in Spanish. This study adheres to these conditions and will properly credit EMT Madrid.

Word count: 83

To ensure a fair and accurate comparison, the analysis focuses only on data from the month of August 2022 for both Dublin and Madrid, providing a controlled timeframe for evaluating bike-sharing usage patterns between the two cities.

Word count: 37

### Comments Dataset Description

The study uses data extracted from Reddit comments concerning the Dublin bike sharing scheme. The comments were collected using Reddit’s API, providing valuable insights into public opinion related to bike-sharing in Dublin.

Word count: 32

### Comments Dataset Licensing and Permissions

The data retrieved from Reddit is governed by Reddit's API terms of service. The use of this data is for academic and research purposes within the scope of Reddit’s guidelines.

Word count: 30

# Data Loading and Initial Assessment

The initial phase of the analysis, was focused on the task of loading and conducting an initial assessment of the datasets for Dublin and Madrid. This step is foundational, ensuring that the data is accurately imported and that its structure and content are well-understood before proceeding with more complex analyses.

Word count: 50

## Dublin Dataset

The Dublin dataset, provided in a CSV file, was loaded into Python using the Pandas library. Choosing Pandas for the CSV format is justified due to its ease of use and ability to handle and analyse large datasets efficiently. The initial commands provided a quick insight into the data's nature and quality. For further information on the data loading process, refer to Appendix D.

The dataset has several columns that detail the bike rental information. Below is a snapshot of the structure of the data, which includes the first five records:

A white table with black text

Description automatically generated

Figure : First five rows of Dublin bike rental data for August 2022

Upon loading, commands like bikes\_data\_aug\_2022.describe() and ikes\_data\_aug\_2022.info() were executed to provide an overview of the data, including the range, mean, and count of numerical values, as well as the total number of non-null entries in each column. This step helped identify the data types and any immediate presence of missing values.

A screenshot of a computer

Description automatically generated

Figure : Pandas 'info()' output for Dublin bike rental data

An in-depth outlier analysis was conducted to ensure the integrity of the Dublin and Madrid dataset, with further details and visualizations provided in Appendix A.

Word count: 167

## Madrid Dataset

The Madrid dataset posed a unique challenge with its non-standard zipped JSON format. A specialized approach was used, combining Python’s built-in json and zipfile modules. For a detailed view of the loading process, refer to appendix D.

The same diagnostic steps used for the Dublin dataset were applied to the Madrid dataset. The analysis revealed outliers within the dataset. The decision to keep these outliers allows the analysis to reflect the full spectrum of bike usage in Madrid, capturing the true variability and providing a more accurate picture of user behaviour and system performance.

A screenshot of a table

Description automatically generated

Figure : First five rows of Madrid bike rental data for August 2022

A screenshot of a computer

Description automatically generated

Figure : Pandas ‘info()’ output for Madrid bike rental data

For a detailed exploration of the data and to view the complete set of preliminary commands executed for the Dublin dataset, refer to the Jupyter notebook (Koleva, 2023), specifically the section titled “Dublin Bike Data for the month of August 2022.” Similarly, for the Madrid dataset, refer to the section “Madrid Bike Data for August 2022” within the same notebook.

Word count: 138

# Exploratory Data Analysis

EDA is an essential step in the data science workflow, which summarizes main characteristics, often with visual methods.

To understand the dynamics further, a new metric was introduced. The standard deviation of the capacity ration (available bikes/number of stands) was calculated. This metric highlights stations with significant fluctuations in bike availability, indicating dynamic usage patterns. The method included computing the capacity ratio for each station and then determining its standard deviation.

The findings are presented in a colour-gradient bar chart, illustrating the hourly volatility of the top 5 bike stations. The design adheres to Edwards Tufte’s principles of data visualization, emphasizing clarity, detail and integrity of information (Tufte, 2007). The gradient colour shows the transition from periods of low to high volatility and facilitates the interpretation of the trends.

Word count: 129

## Dublin hourly volatility

A group of graphs showing different colors

Description automatically generated with medium confidence

Figure : Hourly volatility for the top five stations in Dublin

These visualizations reveal that Portobello Road and Portobello Harbour experience significant fluctuations in bike availability, particularly during the morning and evening hours, likely correlating with commuter patterns. In contrast, Heuston Bridge (South) shows elevated volatility during midday, which may suggest a lunchtime rush or tourist activity.

Word count: 46

### Strategies for improvement

To improve the bike-sharing system, a few strategic actions could be taken:

Bike Redistribution: During peak hours, especially at busy stations like Portobello Road, actively redistribute bikes from quieter stations to meet the high demand.

Infrastructure Expansion: For stations with consistently high traffic, such as Portobello Harbour, increasing the number of docks would ensure that users always have access to bikes and parking spaces.

Safety and Visibility Enhancements: In the evenings, particularly at stations like Heuston Bridge (South) which see increased use, enhance safety with better lighting and clear signs to ensure a user-friendly experience.

Word count: 95

## Madrid hourly volatility

A group of graphs showing the amount of volatility

Description automatically generated

Figure : Hourly volatility for the top five stations in Madrid

Similar to Dublin, the Madrid stations display varied peak usage times. For instance, Plaza de Nelson Mandela and Plaza de la Cebada show increased standard deviation in the capacity ratio during late evening hours, indicating a spike in bike usage.

Word count: 40

## Similarities between Dublin and Madrid

Both cities have stations with high variability in bike usage, suggesting that dynamic activity at certain stations is a common characteristic of urban bike-sharing systems. Stations in both cities, like Portobello Road in Dublin and Plaza de Nelson Mandela in Madrid, have elevated activity in the evenings, indicating a trend that could be associated with leisure or nightlife.

The insights from the volatility patterns suggest that both Dublin and Madrid could benefit from similar operational strategies, such as dynamic redistribution of bikes to match the varying demand throughout the day.

Word count: 90

# Inferential Statistics

A comprehensive analysis was conducted to understand the bike-sharing systems in Dublin and Madrid. Several key steps were performed. An in-depth overview of the complete statistical methodology, including data alignment, normality tests, hypothesis testing, and ECDF visualizations, is documented in Appendix B: Statistical Analysis Details.

Word count: 45

## Data preparation and Normality Check

Data from both datasets was aligned for consistent comparison. A normality test was conducted and determined the data did not follow a normal distribution, guiding the selection of non-parametric tests.

Word count: 30

## Sampling

A 5% random sample was extracted from each dataset for manageability while ensuring representativeness using the Empirical Cumulative Distribution Function (ECDF).

Word count: 21

## Confidence Intervals

Estimated the average number of available bike stands in Dublin using a 95% confidence interval, revealing a narrow range that suggests a uniform distribution of available bike stands across the city.

Word count: 31

## Hypothesis Testing

Employed the Mann-Whitney U test to compare the distribution of trips, finding no significant difference in the central tendency of trips between the two cities.

Utilized the Chi-Squared test to investigate the association between the day of the week and city for trips, with results showing no significant association.

Conducted the Kruskal-Wallis H test to explore bike availability across different times of the day, identifying significant variability in Madrid but not in Dublin.

Word count: 73

## Findings

While the distribution of trips might be similar between Dublin and Madrid, and the pattern of bike usage across the week does not significantly differ between the two cities, there are notable differences in bike availability throughout the day in Madrid, unlike Dublin. This could imply differing dynamics or user behaviours in how bikes are utilized at various times in Madrid.

The confidence interval provided a precise estimate of bike stand availability in Dublin, indicating an efficient distribution and management of the bike-sharing infrastructure.

Word count: 107

# Machine Learning Analysis

## Data importation and preliminary setup

The dataset utilized in this analysis is the already cleaned and prepared bike availability data across various stations in Dublin for the month of August 2022. This data was previously processed and analysed in the "Stats and EDA" Jupyter notebook. To ensure continuity and integrity in the analysis, the cleaned dataset was exported and saved in a CSV format. In this machine learning phase, the already pre-processed dataset was imported.

Word count: 70

## Data Preparation and Feature Development

Initially, the ‘time’ data was converted to a format suitable for detailed time-based analysis. This helped in gathering data about bike availability at different times across all stations, giving a clear picture of trends in bike usage. The analysis was then improved by adding new time-related features like hours and days, which were extracted from the ‘time’ column.

The dataset, consisting of 1488 data points, was split to support the modelling process. In line with standard time series analysis procedures recommended by Hyndman and Athanasopoulos (2018), the initial 1100 data points - representing roughly 80% of the dataset - were utilized to train the model. The subsequent portion of the dataset was allocated for testing purposes. This split ensures that the model is trained on historical data and assessed on the most current data, thereby copying a real-life prediction situation.

Word count: 140

## Model Selection and Refinement

In the study analysing bike availability in Dublin, the selection of both linear and non-linear models was crucial for a detailed understanding of the data. These models are part of supervised learning, which means they learn from known data to make predictions.

Word count: 42

### Linear Models with Feature Engineering

Linear Regression was chosen for its basic but powerful ability to predict outcomes by identifying straight-line relationships between variables. According to 'An Introduction to Statistical Learning' (James et al., 2013), it predicts a response (Y) based on the predictor (X) and is expressed simply as Y ≈ β0 + β1X. To make Linear Regression work well with categories in the data, OneHotEncoder was used to turn those categories into numbers the model can understand.

Ridge Regression was selected to improve upon Linear Regression, especially in cases where the data points are too closely related (multicollinearity). By introducing Polynomial Features, Ridge Regression could also understand curves in the data, making it more versatile.

Word count: 112

### Non-Linear Model – RandomForestRegressor

The RandomForestRegressor, a more complex tool, was used to look into patterns that aren't straight lines. It's a type of supervised learning too and works by combining multiple smaller models to get a more accurate and robust understanding.

Further details and the impact of these feature engineering techniques on model performance are discussed in Appendix C: Machine Learning Analysis Details.

Word count: 60

## Evaluation

In this section, the performance of the predictive models is assessed using key statistical measures, which are presented visually for comparison. The R-squared (R²) score, Mean Absolute Error (MAE), and Mean Squared Error (MSE) are the chosen metrics. The R² score indicates how well the model explains the variation in the data, the MAE reflects the average error in predictions, and the MSE measures the average of the errors squared.

Figure 7 displays these metrics for each model. The Ridge Regression model with Polynomial Features shows the highest R² score, suggesting it predicts bike availability with the greatest accuracy among the models tested. It also has the lowest MSE, implying it makes smaller errors on average.

The Random Forest model's R² score is slightly lower, but its MAE is almost as good as that of the Ridge Regression model, indicating it is also a strong predictor.

The Linear Regression model, while straightforward, has a lower R² score and higher error metrics. This suggests it does not predict as accurately as the other models.

A graph with a bar

Description automatically generated with medium confidence

Figure : Evaluation of the models

Adhering to Tufte's principles of graphical integrity and clarity, the visualizations in Figure 7 are designed to present complex statistical information in a clear, precise, and efficient manner.

Word count: 201

## Decision Tree and Random Forest

Additionally, Decision Tree and Random Forest classifiers were employed after creating a target variable for bike usage. A modest improvement in prediction accuracy after hyperparameter tuning for the Decision Tree was observed. For in detail analysis, refer to ‘ML-bikedata’ notebook in ‘Decision Tree and Random Forest’ markdown.

Word count: 45

# Sentiment Analysis

To analyse the public sentiment towards Dublin Bikes, a sentiment analysis was conducted using comments extracted from a Reddit post titled 'How are Dublin Bikes these days?'.

The praw library was used to interface with the Reddit API, adhering to best practices by storing sensitive credentials in a separate JSON file—a measure ensuring security and adherence to software development protocols. For more details in regards to pulling the comments, refer to Appendix C5.

The comments were first pre-processed to remove noise such as URLs, special characters, and numbers, and then tokenized into individual words. Stopwords were removed to focus on the more meaningful content of the comments. Using the nltk library's SentimentIntensityAnalyzer, each comment was analyzed to determine the sentiment scores, which were further categorized into positive, negative, or neutral sentiments based on a threshold set according to standard guidelines for sentiment analysis tools, (Hutto and Gilbert, 2014).

The sentiment analysis indicated that positive reactions were most common among the dataset's comments, alongside a notable amount of negative and neutral opinions. A bar chart that visually summarizes the distribution of sentiments a shown in Figure 8. For a visual representation of the most frequent terms found in the comments, refer to the word cloud in Appendix C5.

A graph with different colored squares

Description automatically generated

Figure : Sentiment distribution of comments

Also, a machine learning model using the Bag of Words approach was trained to predict the sentiment of comments. The model achieved an accuracy score of 62%. Considering the limited size of the data, the accuracy score is a good baseline for further model refinement.

Word count: 261

## Interactive Dashboard

An interactive dashboard was created using Dash, a Python web application framework. The dynamic dashboard visualizes Dublin’s bike availability, using different visualizations such as time-series analysis, heatmaps, and geographical mapping. The visualizations adhere to Edward Tufte’s principles of design, maximizing data density and integrity and presenting the information in a clear and efficient way. To view the visualization, refer to the Jupyter Notebook titled ‘ML-bikedata’.

Word count: 65

# Conclusion and Recommendations

The analysis of bike-sharing systems in Dublin and Madrid gave insights into its dynamics and user preferences. While there are some distinct usage patterns, there are also similarities and areas for improvement.

In both Dublin and Madrid high variability in bike usage was observed in certain stations, particularly during peak hours. To address this, a dynamic system of redistribution can be implemented, moving bikes from quieter stations to busier ones in order to meet high demand. This approach would require the use of forecasting models, similar to the one implemented in this study.

For stations that experience usually high traffic, increasing the number of available docks and bikes is crucial. Also, optimizing the location of new stations based on usage patterns can improve the system’s accessibility and convenience, which will improve the user satisfaction.

The sentiment analysis of comments about Dublin’s bike-sharing system showed a mixture of positive, negative, and neutral comments. Even though the positive comments were the highest number, there is need for service improvement. A good starting point would be to consciously monitor and analyse public feedback.

In conclusion, while Dublin and Madrid have their unique contexts, they share common goals and challenges in bike-sharing. By using data-driven strategies, engraining with the public, and continuously adapting to the changing dynamics, both cities can improve their bike-sharing systems to meet the citizens needs and move towards a more sustainable and mobile future.

Word count: 234

Total word count: 3025

# Appendix A: Outlier Analysis

## A1: IQR Visualization for Dublin Dataset

The IQR is the range between the first quartile (25th percentile) and the third quartile (75th percentile) and is used to measure the spread of the middle 50% of the data. Outliers are typically identified as observations that fall below Q1 - 1.5IQR or above Q3 + 1.5IQR.

A blue rectangular box with black lines

Description automatically generated with medium confidence

Figure : IQR analysis for Dublin

The absence of data points beyond the whiskers in the Dublin dataset indicates no significant outliers, suggesting the data's consistency and reliability.

## A2: IQR Visualization for Madrid Dataset

The boxplot below represents the distribution of the primary variable of interest in the Madrid dataset.

A graph with a blue line

Description automatically generated with medium confidence

Figure : IQR analysis for Madrid

The outlier analysis for the Madrid dataset, conducted using the Interquartile Range (IQR) approach, revealed several data points outside the typical range. These outliers were not attributed to data entry errors but rather to the inherent dynamics of bike usage patterns, such as peak usage times and the popularity of certain stations.

Given the focus of the analysis on the overall bike usage, including its extremes, these outliers were intentionally kept.

# Appendix B: Statistical Analysis Details

## B1: Data Alignment and pre-processing

The datasets for Dublin and Madrid were standardized to ensure compatibility for comparative analysis. This involved renaming columns, converting timestamps to datetime objects to facilitate time series analysis, and aligning datasets to the same columns for direct comparison.

dublin\_data\_renamed = bike\_usage.rename(columns={

'name': 'station\_name',

'time': 'timestamp',

'bike\_stands': 'total\_stands',

'available\_bike\_stands': 'available\_stands',

'available\_bikes': 'available\_bikes'

})

dublin\_data\_renamed['timestamp'] = pd.to\_datetime(dublin\_data\_renamed['timestamp'])

madrid\_data\_renamed = bike\_usage\_madrid.rename(columns={

'name': 'station\_name',

'\_id': 'timestamp',

'total\_bases': 'total\_stands',

'free\_bases': 'available\_stands',

'available\_bikes': 'available\_bikes'

})

madrid\_data\_renamed['timestamp'] = pd.to\_datetime(madrid\_data\_renamed['timestamp'])

## B2: Sampling Techniques

Sampling was performed to make the data analysis more computationally feasible. A 5% random sample from each dataset was used, ensuring representativeness using the ECDF.

Random sampling is used because it minimizes bias and makes it more likely that the sample will accurately reflect the structure of the population. The choice of a 5% fraction was a balance between having a sample size large enough to retain the statistical characteristics of the full datasets, while being small enough to allow for faster computation.

# For the Dublin aligned dataset

dublin\_data\_sample = dublin\_data\_renamed.sample(frac=0.05, random\_state=42)

dublin\_data\_sample.reset\_index(drop=True, inplace=True)

# For the Madrid aligned dataset

madrid\_data\_sample = madrid\_data\_renamed.sample(frac=0.05, random\_state=42)

madrid\_data\_sample.reset\_index(drop=True, inplace=True)

By plotting the ECDF of the sample data against the ECDF of the full data, a visual confirmation can be seen of the representativeness of the sample. In the plots, if the sample ECDF closely follows the ECDF of the full data, this indicates that the sample is a good representation of the full dataset. The plots for this project showed that the samples for both cities closely mirrored the full datasets, confirming the effectiveness of the sampling method used.

The plots visually demonstrate the similarity in distribution between the samples and the full datasets.

A graph of a graph of a number of different types of graphs

Description automatically generated with medium confidence

Figure : ECDF plot

# Appendix C: Machine Learning Analysis Details

## C1: Code for pre-processing and Model Evaluation

The pre-processing stage involved converting timestamps to a uniform numerical format suitable for machine learning models. The POSIX time format was chosen for its compatibility with various regression models.

# convert to POSIX time by dividing by 10\*\*9

X = available\_bikes\_series\_22.index.astype("int64").values.reshape(-1, 1) // 10\*\*9

## C2: Feature Engineering Techniques

OneHot Encoding

Categorical features, like hours and days of the week, were initially in a format not directly suitable for regression analysis. OneHot encoding was used to convert these categorical variables into a numerical format that can be provided to the linear models, allowing for a more nuanced representation that captures the unique effect of each category.

enc = OneHotEncoder()

X\_day\_hour\_week\_onehot = enc.fit\_transform(X\_day\_hour\_week).toarray()

X\_day\_hour\_week\_onehot

Adding Polynomial Features

Polynomial features allow the model to capture more complex relationships by considering not only the individual features but also their interactions and powers up to a specified degree. This is particularly useful for linear models to fit non-linear patterns.

poly\_transformer = PolynomialFeatures(degree = 2, interaction\_only = True, include\_bias = False)

X\_day\_hour\_week\_onehot\_poly = poly\_transformer.fit\_transform(X\_day\_hour\_week\_onehot)

lr\_poly = Ridge()

eval\_on\_features(X\_day\_hour\_week\_onehot\_poly, y, lr\_poly, plot=True)

## C3: Impact on Model Performance

The initial models, which used the POSIX time as the sole feature, performed poorly, with an R^2 score near zero for the RandomForestRegressor. This indicated that the models failed to capture any meaningful relationship between time and bike availability. After applying OneHot encoding to the categorical features, a significant improvement was observed in the model's performance. The RandomForestRegressor's R^2 score increased to 0.73, indicating a good fit to the data. Further improvements were achieved by adding polynomial features. The Ridge regression model with polynomial features outperformed the simpler models, getting an R^2 score of 0.79. This suggests that the additional complexity allowed the model to uncover more subtle patterns in the data.

## C4: Visualization of Model Performance

For all visualizations in this section the blue line represents the actual bike rentals for the training set, while the orange line shows the actual rentals for the test set. The green dashed line shows the model's predictions for the training set, and the red dashed line shows the predictions for the test set.

This figure illustrates the predictive performance of the Random Forest Regressor with features extracted from the day of the week and the hour of the day. The close alignment of the red dashed line with the orange test line indicates the model's effectiveness in capturing the pattern of bike rentals over time.

A graph showing the time of a wave

Description automatically generated with medium confidence

Figure : Performance of Random Forest Model with Day and Hour Features

The following shows the performance of the Linear Regression with the OneHot encoder applied.

A graph showing the time of a week

Description automatically generated with medium confidence

Figure : Performance of Linear Regression Model with OneHot Encoded Features

A visualization showing the performance of the Ridge regression with polynomial features.

A graph showing a number of data

Description automatically generated with medium confidence

Figure : Performance of Ridge model with Polynomial features

## C5: Reddit API call details

This script is designed to use the Python Reddit API Wrapper (PRAW) to collect comments from a specific Reddit submission (post) and save them into a CSV file.

# Initialize PRAW

reddit = praw.Reddit(client\_id=client\_id, client\_secret=client\_secret, user\_agent=user\_agent)

submission\_id = '12pqv76'

csv\_file\_path = 'reddit\_bike\_comments.csv'

fieldnames = ['comment', 'created\_utc', 'submission\_title', 'submission\_id']

with open(csv\_file\_path, mode='w', newline='', encoding='utf-8') as file:

writer = csv.DictWriter(file, fieldnames=fieldnames)

writer.writeheader()

submission = reddit.submission(id=submission\_id)

submission.comments.replace\_more(limit=None)

for comment in submission.comments.list():

writer.writerow({

'comment': comment.body,

'created\_utc': datetime.datetime.fromtimestamp(comment.created\_utc),

'submission\_title': submission.title,

'submission\_id': submission.id,

})

print("Data collection complete. Check the CSV file called 'reddit\_bike\_comments'.")

Initialization:

Sets up the PRAW instance with necessary credentials (client\_id, client\_secret, user\_agent). These credentials are obtained when registering a script as an app on Reddit's developer site. For detailed instructions and documentation on setting up PRAW, refer to the PRAW documentation (PRAW, 2024).

Variables Setup:

submission\_id: The unique identifier for the Reddit post from which comments are to be fetched.

csv\_file\_path: The path where the CSV file containing the comments will be saved. fieldnames: Headers for the CSV file, including 'comment', 'created\_utc', 'submission\_title', and 'submission\_id'.

CSV Preparation:

Opens a new CSV file at the specified path and writes the headers to it. This sets up the file to store the comments and their associated data.

Extract and Save Comments: Iterates through all comments in the submission. For each comment, the script extracts the comment's text, its creation time (converted to a readable format), the title of the submission, and the submission ID. This information is then written as a new row in the CSV file.

## C6: Word cloud visualization for sentiment analysis

A close up of words

Description automatically generated

Figure : Word cloud of sentiments from Dublin Bikes Reddit Comments

# Appendix D: Data Loading

## D1: Dublin Dataset

The following code snippet illustrates the process:

*# Load the dataset*

bikes\_data\_aug\_2022\_dublin **=** pd**.**read\_csv("dublinbike-historical-data-2022-08.csv")

*# Display the first five rows*

bikes\_data\_aug\_2022\_dublin**.**head()

## D2: Madrid Dataset

The process began with extracting the JSON file from the zipped archive and then parsing it line by line due to its non-standard structure:

*# Path to the zip file and directory for extraction*

zip\_path **=** “C:/Users/mariy/CA2/202208-json.zip”

extract\_dir **=** “C:/Users/mariy/CA2/extracted\_json”

*# Unzip the file*

**with** zipfile**.**ZipFile(zip\_path, ‘r’) **as** zip\_ref:

zip\_ref**.**extractall(extract\_dir)

*# Get the path to the JSON file*

json\_file\_path **=** os**.**path**.**join(extract\_dir, ‘202208.json’)

*# Function to parse JSON line by line*

**def** parse\_json\_line\_by\_line(file\_path):

data **=** []

**with** open(file\_path, ‘r’, encoding**=**’utf-8’) **as** file:

**for** line **in** file:

**try**:

json\_object **=** json**.**loads(line)

data**.**append(json\_object)

**except** json**.**JSONDecodeError **as** e:

print(f”Error parsing line: {e}”)

**return** data

*# Parse the JSON file*

line\_by\_line\_data **=** parse\_json\_line\_by\_line(json\_file\_path)

The dataset contained nested ‘stations’ data within each line. A loop was employed to iterate through the parsed data, extract and flatten this nested information, and then compile it into a comprehensive DataFrame for further analysis:

*# Flatten the nested “stations” data*

flattened\_stations **=** pd**.**DataFrame()

**for** index, row **in** pd**.**DataFrame(line\_by\_line\_data)**.**iterrows():

stations\_df **=** pd**.**DataFrame(row[‘stations’])

stations\_df[‘\_id’] **=** row[‘\_id’]

flattened\_stations **=** flattened\_stations**.**append(stations\_df, ignore\_index**=True**)

*# Display the first few rows of the flattened DataFrame*

print(flattened\_stations**.**head())

The unconventional structure of the Madrid dataset necessitated a custom script. The json module’s ability to parse JSON line by line was crucial given the file’s format, and the iterative flattening process ensured that the nested data was transformed into a format suitable for comparative analysis with the Dublin data. To facilitate easier access and manipulation in the analyses, the cleaned and transformed DataFrame was saved as a new CSV file. This step ensures that the data can be quickly reloaded without the need to repeat the pre-processing steps, thus optimizing the efficiency of the workflow.

# References

Dublin City Council. (2022). The Dublin City Development Plan 2022-2028, Chapter 8: Sustainable Movement and Transport. Available at: https://www.dublincity.ie/sites/default/files/2022-12/Final%201-08%20Sus%20Mov.pdf (Accessed: 28 December 2023)

City of Madrid. (2022). Recovery, Transformation and Resilience Plan for the City of Madrid. Available at: https://www.madrid.es/UnidadWeb/NxC/PlanRecuperacion/rtsingles.pdf (Accessed: 28 December 2023).

Dublin City Council. (2022). Dublinbikes API. Available at: https://data.gov.ie/dataset/dublinbikes-api (Accessed: 1 December 2023)

Creative Commons. (2021). Creative Commons — Attribution 4.0 International — CC BY 4.0. Available at: https://creativecommons.org/licenses/by/4.0/ (Accessed: 28 December 2023)

EMT Madrid. (2022). Datos generales. Available at: https://opendata.emtmadrid.es/Datos-estaticos/Datos-generales-(1) (Accessed: 12 December 2023)

EMT Madrid. (2022). EMT Madrid Open Data. Available at: https://opendata.emtmadrid.es/Documentos/terminosycondiciones.aspx (Accessed: 29December 2023)

Koleva, M. (2023). EDA and Stats - bikedata. Available at: https://github.com/mkoleva0/DataAnalytics\_CA2/blob/main/EDA%20and%20Stats%20-%20bikedata.ipynb (Accessed: 28 January 2024)

Tufte, E. (2007) The Visual Display of Quantitative Information (2nd ed.). Cheshire, Connecticut: Graphics Press

James, G., Witten, D., Hastie, T. and Tibshirani, R., 2013. An Introduction to Statistical Learning. New York: Springer

Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: Principles and Practice (2nd ed.). Monash University. Available at: https://otexts.com/fpp2/accuracy.html (Accessed: 3 January 2024)

PRAW, 2024. Configuration. Python Reddit API Wrapper Documentation. Available at: https://praw.readthedocs.io/en/stable/getting\_started/configuration.html (Accessed 22 December 2023)

Hutto, C.J. and Gilbert, E. (2014) 'VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text', Proceedings of the International AAAI Conference on Web and Social Media, 8(1). Available at: https://ojs.aaai.org/index.php/ICWSM/article/view/14550/14399 (Accessed: 2 January 2024).