**CCT College Dublin**

**Assessment Cover Page**

*To be provided separately as a word doc for students to include with every submission*

|  |  |
| --- | --- |
| **Module Title:** | *Programming for DA*  *Statistics for Data Analytics*  *Machine Learning for Data Analysis*  *Data Preparation & Visualisation* |
| **Assessment Title:** | *MSC\_DA\_Repeat* |
| **Lecturer Name:** | *Marina Iantorno/Taufique Ahmed*  *Sam Weiss*  *Muhammad Iqbal*  *David McQuaid* |
| **Student Full Name:** | Ashish Royan Sanctis  <https://github.com/AshishSanctis/Repeat-SEM-1/upload/main> |
| **Student Number:** | 2023444 |
| **Assessment Due Date:** | 02/08/2024 |
| **Date of Submission:** | 07/08/2024 |

A close-up of a signature

Description automatically generated

**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. |

Ashish Sanctis

Comparative analysis of bikes usage between Dublin and New York City

Table of Contents

[**1.** **Introduction** 3](#_Toc173896905)

[**2.** **Data Acquired Source** 3](#_Toc173896906)

[**2.1 Data source and license** 3](#_Toc173896907)

[**2.2 Dataset Description:** 4](#_Toc173896908)

[**2.2.1 Dublin bikes** 4](#_Toc173896909)

[**2.2.2 New York bikes** 4](#_Toc173896910)

[**3.** **Exploratory Data Analysis** 5](#_Toc173896911)

[3.1 EDA for Dublin bikes (Dublin\_bike\_Analysis.ipynb) 5](#_Toc173896912)

[3.2 EDA for New York bikes (Newyork\_bike\_Analysis.ipynb). 6](#_Toc173896913)

[**4.** **Data Visualization** 8](#_Toc173896914)

[**4.1** **Analysis for Dublin Bikes:** 8](#_Toc173896915)

[**4.1.1 Bar Plot** 8](#_Toc173896916)

[**4.1.2 Line Plot** 9](#_Toc173896917)

[**4. 1.3 Stacked bar plot** 9](#_Toc173896918)

[**4.1.4 Scatter Plot on a Geographical Dublin Map.** 9](#_Toc173896919)

[**4.2** **Analysis for New York bikes:** 10](#_Toc173896920)

[4.2.1 Line graph: 10](#_Toc173896921)

[4.2.2 Count Plot: 10](#_Toc173896922)

[4.2.3 Bar plot: 10](#_Toc173896923)

[4.2.4 Scatter plot on geographic New York map: 10](#_Toc173896924)

[**5.** **Dash for Visualization** 11](#_Toc173896925)

[**6.** **Data Pre-processing for Dublin\_bikes.** 13](#_Toc173896926)

[**6.1** **Feature Extraction:** 13](#_Toc173896927)

[**6.2 Feature Selection:** 13](#_Toc173896928)

[**6.3** **Scaling:** 14](#_Toc173896929)

[**7.** **Machine Learning** 15](#_Toc173896930)

[**7.1 Linear Regression:** 15](#_Toc173896931)

[**7.2 Comparative Analysis of different ML models used:** 15](#_Toc173896932)

[**7.3 Hyperparameter Tuning:** 16](#_Toc173896933)

[**7.4 Generalization using Cross validation of ML models:** 17](#_Toc173896934)

[**8.** **Machine Learning for newyork\_bikes dataset (Newyork\_bike\_usage.ipynb)** 17](#_Toc173896935)

[8.1 Dash to visualize machine learning model results. 18](#_Toc173896936)

[**9.** **Programming for Data Analytics** 19](#_Toc173896937)

[Reference 21](#_Toc173896938)

**Table of Tables**

[Table 1: Dataset description for Dublin\_bikes 6](#_Toc173896224)

[Table 2: Dataset description for New York bikes 6](#_Toc173896225)

[Table 3: Skewness values of variables in dublin\_bikes 7](#_Toc173896226)

[Table 4: Dataset features and description 8](#_Toc173896227)

[Table 5: Comparison of different ML models 17](#_Toc173896228)

[Table 6: Comparison of different ML models after hyperparameter tuning 17](#_Toc173896229)

[Table 7: Cross validation for generalization of different models used in analysis 18](#_Toc173896230)

[Table 8: The dataset used for ML models ‘newyork\_bikes\_corr’’ 18](#_Toc173896231)

[Table 9: Comparative Analysis of Machine learning models 19](#_Toc173896232)

**Table of Figures**

[Figure 1 Dash to visualize number of bikes available in Dublin 14](#_Toc173896233)

[Figure 2: Dash to visualize NewYork Bikes availability. 15](#_Toc173896234)

[*Figure 3: Correlation between variables* 15](#_Toc173896235)

[Figure 4: Selected features for Machine learning model – dataset ‘dublin\_bikes\_corr’ 16](#_Toc173896236)

[Figure 5: Dash for machine learning models 20](#_Toc173896237)

# **Introduction**

A growing global trend towards sustainable and healthful transportation is reflected in the fact that cycling has become an essential aspect of urban life. Bicycle infrastructure is increasing significantly in Irish cities like Cork and Dublin, where public bike-sharing programs and bike lanes are encouraging more people to travel by bicycle. This change not only encourages physical health but also lessens the impact on the environment and traffic congestion. (cyclingireland, n.d.)

Similar to other cities, New York City has accepted cycling as a practical form of transportation, as evidenced by the vast networks of bike lanes and the well-liked Citi Bike initiative. The city's ongoing investments in cycling infrastructure and community programs demonstrate its dedication to creating a bike-friendly environment. Ireland and New York both provide as examples of how metropolitan areas can change to meet the demands of contemporary transportation while emphasizing the advantages of cycling for both people and the larger community.

This report focuses on the analysis of bikes availability and usage at various locations in Dublin and New York based on the data available for the month of June 2024.The analysis involves a combination of exploratory data analysis (EDA), data visualization, data preprocessing and the implementation of various machine learning models. The objective is to understand the trends of bike usage and bike availability across Dublin and New York.

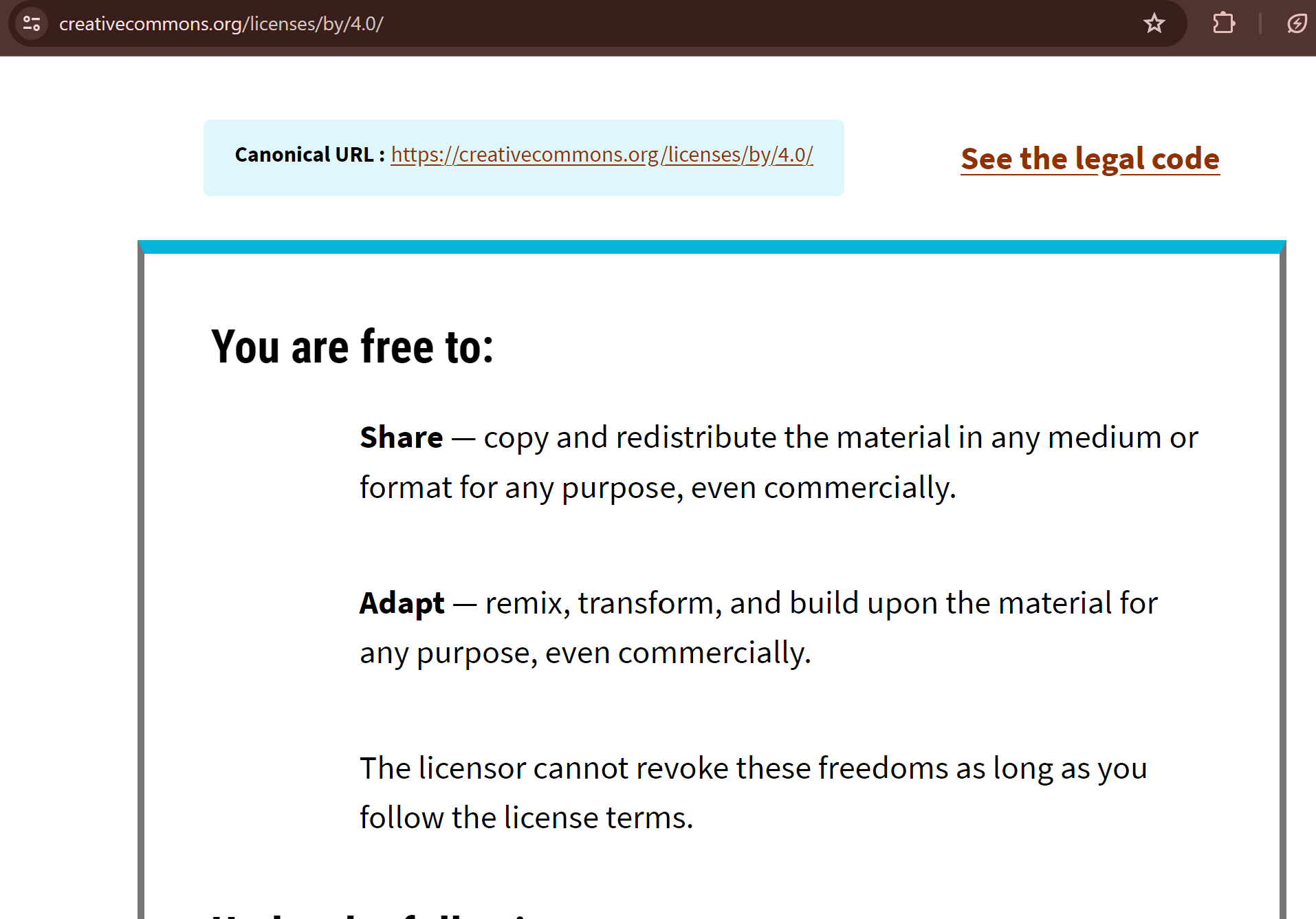
# **Data Acquired Source**

## **2.1 Data source and license**

The data for Dublin bikes is sourced from <https://data.gov.ie/dataset/dublinbikes-api>.

License details can be found at: [**Creative Commons Attribution 4.0**](https://creativecommons.org/licenses/by/4.0/) or https://opendefinition.org/licenses/cc-by

<https://data.gov.ie/dataset/dublinbikes-api/resource/9496fac5-e4d7-4ae9-a49a-217c7c4e83d9>



For New York city bike, the data is explored from [Citi Bike System Data | Citi Bike NYC](https://ride.citibikenyc.com/system-data) and dataset can be downloaded at [Index of bucket "tripdata"](https://s3.amazonaws.com/tripdata/index.html). A zip file ‘202406-citibike-tripdata’ is downloaded containing data for June 2024.

License is available at - <https://ride.citibikenyc.com/data-sharing-policy>

*“Lyft Bikes and Scooters, LLC (“Bikeshare”) operates New York City’s Citi Bike bicycle sharing service. Bikeshare is committed to supporting bicycling as an alternative transportation option. As part of that commitment, Bikeshare makes certain Citi Bike system data (“Data”) available to the public, subject to the terms and conditions of this License Agreement (“Agreement”). By accessing or using any of the Data, you agree to all of the terms and conditions of this Agreement.”*

**1. License.** Bikeshare hereby grants to you a non-exclusive, royalty-free, limited, perpetual license to access, reproduce, analyze, copy, modify, distribute in your product or service and use the Data for any lawful purpose (“License”).

## **2.2 Dataset Description:**

For comparative analysis, I have used two separate datasets ‘Dublin\_bikes’ created from “dublin-bikes\_station\_status\_062024.csv” and ‘newyork\_bikes’ created from “202406-citibike-tripdata\_1.csv” using the pandas library pd and function read\_csv(). The datasets are analysed in separate jupyter notebooks , ‘Dublin\_bike\_Analysis.ipynb’ and ‘Newyork\_bike\_Analysis.ipynb’. Since not all of the attributes in each dataset are the same, performing a separate analysis allows for a better understanding and insights of the data from each city. Combining the datasets would result in a loss of the complete overview. However, data is collected for the same time period, from June 1 to June 15, 2024, for comparison study.

The dataset details and description for Dublin\_bikes and newyork\_bikes are provided in below section.

### **2.2.1 Dublin bikes**

The dataframe ‘dublin\_bikes’ consist of information on the status of bike-sharing stations, which can be useful for users looking to find available bikes or docks, and for city planners to monitor and optimize the bike-sharing infrastructure. It includes columns system\_id, last\_reported, station\_id, num\_bikes\_available, num\_docks\_available, is\_installed, is\_renting, is\_returning, name, short\_name, address, lat, lon, region\_id and capacity.

The dataset consists of categorical columns system\_id, is\_installed, is\_renting, is\_returning, name, short\_name, address and continuous/numeric columns station\_id, last\_reported, num\_bikes\_available, num\_docks\_available, lat, lon, region\_id and capacity which are continuous in nature.

|  |  |
| --- | --- |
| **Column Name** | **Description** |
| **system\_id** | The bike-sharing system's identifier. Dublin Bikes is identified as the system in this dataset by the value "dublin\_bikes". |
| **last\_reported** | A timestamp (YYYY-MM-DD HH:MM) indicating when the station's data was last reported. |
|  |
| **station\_id** | A special number assigned to every bike station. |  |
| **num\_bikes\_available** | The total number of bikes that can be rented at the station right now. |  |
| **num\_docks\_available** | The number of docks that are open at the station right now for bikes that are being returned. |  |
| **is\_installed** | Boolean value indicating whether the station is installed (True or False). |  |
| **is\_renting** | Boolean value indicating whether bikes are being rented out at the station at the moment (True or False). |  |
| **is\_returning** | Boolean value indicating whether returns are being accepted by the station at this time (True or False). |  |
| **name** | The bike station's name. |  |
| **short\_name** | This section has NaN values in the dataset. |  |
| **address** | The address of the bike station. |  |
| **lat** | Latitude coordinate of the station location. |  |
| **lon** | Longitude coordinate of the station location. |  |
| **region\_id** | This section has NaN values in the dataset. |  |
| **capacity** | Total capacity of the station, indicating the total number of bike docks (sum of available bikes and docks). |  |

Table 1: Dataset description for Dublin\_bikes

### **2.2.2 New York bikes**

The dataset ‘newyork\_bikes’ includes rider-specific data from a citi bike New York. Every record is a single ride and contains information on the start and finish locations, the kind of bike that was used, and the user category. The dataset is used to examine user behaviour, station popularity, and bike utilization trends and bike ride duration at various stations in New York.

|  |  |
| --- | --- |
| **Column Name** | **Description** |
| **ride\_id** | Unique identifier for each bike ride. |
| **rideable\_type** | Type of bike used for the ride (e.g., electric\_bike, classic\_bike). |
| **started\_at** | Timestamp (YYYY-MM-DD HH:MM.SSS) indicating when the ride started. |
|  |
| **ended\_at** | Timestamp (YYYY-MM-DD HH:MM.SSS) indicating when the ride ended. |  |
|  |
| **start\_station\_name** | Name of the station where the ride started. |  |
| **start\_station\_id** | Identifier for the station where the ride started. |  |
| **end\_station\_name** | Name of the station where the ride ended. |  |
| **end\_station\_id** | Identifier for the station where the ride ended. |  |
| **start\_lat** | Latitude coordinate of the start station. |  |
| **start\_lng** | Longitude coordinate of the start station. |  |
| **end\_lat** | Latitude coordinate of the end station. |  |
| **end\_lng** | Longitude coordinate of the end station. |  |
| **member\_casual** | Type of user: 'member' for registered users or 'casual' for non-registered users. |  |

Table 2: Dataset description for New York bikes

# **Exploratory Data Analysis**

## EDA for Dublin bikes (Dublin\_bike\_Analysis.ipynb)

Read the first and last five observations of dataset ‘dublin\_bikes’, and observed the column ‘region\_id’ and ‘short\_name’ have null values. Identified the shape of the dataset using .shape function and we have 315747 observations and 15 features/variables. The .info() is used to identify the data types of each variables and non-null values. It is observed that data type for variables is as expected but variables ‘short\_name’ and ‘region\_id’ have zero non-null values. Same is validated using .count() and .isnull().sum() function which shows 315747 observations are missing for these columns and hence dropped these columns. After dropping the columns, we have 315747 rows and 13 columns. Identified the unique values for 'system\_id', 'name', 'address' using .unique() function and we identified that we have data for 114 different bike station in Dublin. A check for duplicate rows is done using . duplicated() function.

Converted the variable ‘last\_reported’ datatype to datetime64 format for further analysis. Renamed the columns "last\_reported" as "time\_stamp" and "address" as "location" for better understanding. Dropped the column "name" as the data is identical in ‘name’ and ‘location’ column, the only difference being upper case and lower case. Verifying for are any invalid values in latitude and longitude observations by checking the ‘lat’ and ‘lon’ observations lie in the range (-90 to 90) and (-180 to 180) respectively.

* + 1. Feature engineering and transformation:

Added two more columns ‘hour’, ‘day\_of\_week’ by extracting values from the ‘time\_stamp’ variable by using function .dt.hour() and .dt.dayofweek.

A new feature ‘utilization’ is derived from existing data, for further analyse the number of bikes available at different station at specific hour, day of the week, weekend and weekdays. These features are added to provide additional insights and improve the predictive power of machine learning models.

station utilization is calculated by using formula:

**‘utilization’ = [(‘num\_bikes\_available’)/ (‘capacity’) ]\* 100**

* + 1. Box Plot to identify outliers and skewness:

An Outlier is an observation in a given dataset that lies far from the rest of the observations. (analyticsvidhya, n.d.) The box plot does not show any outliers for any of the variables in the dataset, which indicates there are no extreme data points in the distribution.

The box plot and displot is used to visualize the outliers and data distribution of all the variables, to get an insight into the dataset. (Refer section 3.5 and 3.6 in Dublin\_bike\_Analysis.ipynb)

|  |  |  |
| --- | --- | --- |
| **Variables** | **Skewness value** | **Description** |
| station\_id | 0.054176 | Very close to 0, indicating a nearly symmetrical distribution. |
| num\_bikes\_available | 0.75603 | Moderately skewed to the right (positive skew). |
| num\_docks\_available | -0.04233 | Close to 0, indicating a nearly symmetrical distribution. |
| lat | 0.095911 | Very slightly skewed to the right. |
| lon | -0.47196 | Moderately skewed to the left (negative skew). |
| capacity | -0.39732 | Moderately skewed to the left. |
| utilization | 0.512813 | Moderately skewed to the right. |
| hour | -0.08211 | Close to 0, indicating a nearly symmetrical distribution. |
| day\_of\_week | -0.18434 | Slightly skewed to the left. |

Table 3: Skewness values of variables in dublin\_bikes

station\_id, num\_docks\_available, hour, have the skewness value close to zero, which indicates a symmetrical data distribution.

num\_bikes\_available, utilization, have moderate positive skewness values (0.75 and 0.51). This indicates, most stations have moderate number of bikes available, while few stations have significantly more number of bikes available.

lon, capacity: have moderate negative skewness values, it means that most places are spread out on the right side of the longitude spectrum. It suggests that while most stations have a high capacity, some have capacities that are noticeably lower.

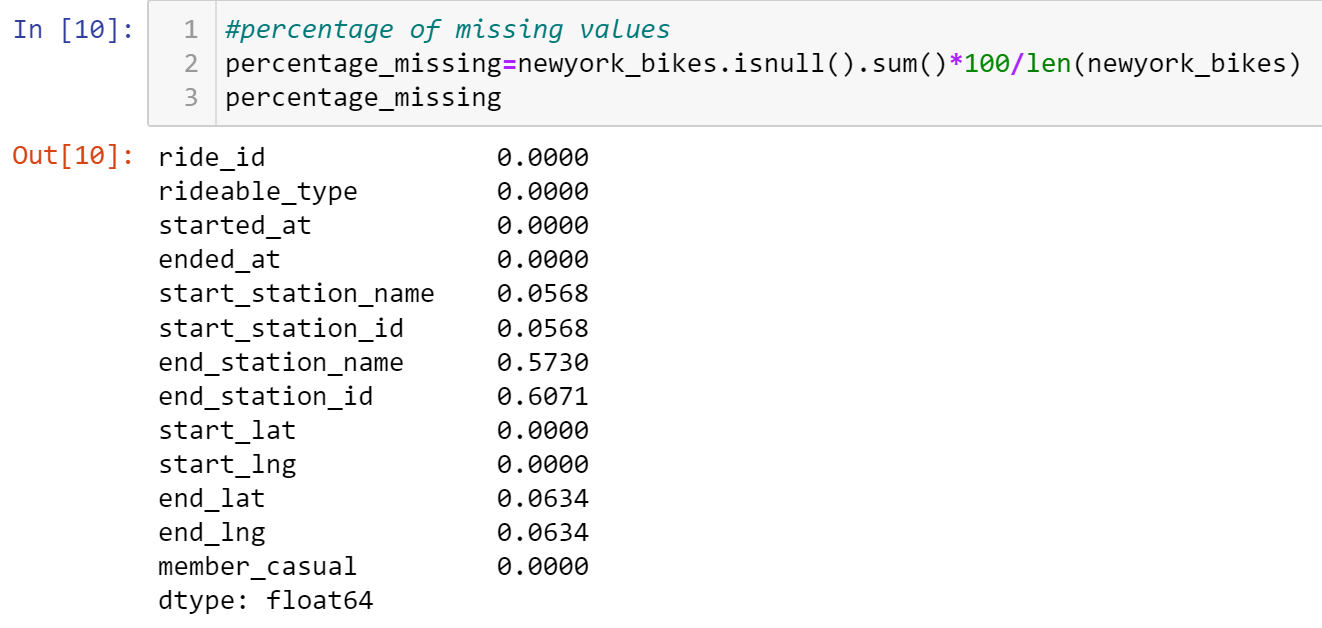
## 3.2 EDA for New York bikes (Newyork\_bike\_Analysis.ipynb).

The dataset ‘newyork\_bikes’ has a shape of (1000000, 13), the detailed description is mentioned below in Table 4.

|  |  |
| --- | --- |
| **Feature** | **Description** |
| **ride\_id** | Unique ID for each ride. |
| **rideable\_type** | Type of bike used for the ride, e.g., electric\_bike or classic\_bike. |
| **started\_at** | Date and time when the ride began, formatted as YYYY-MM-DD HH:MM |
|  |
| **ended\_at** | Date and time when the ride ended, formatted as YYYY-MM-DD HH:MM |  |
|  |
| **start\_station\_name** | Name of the station where the ride started. |  |
| **start\_station\_id** | ride start station id |  |
| **end\_station\_name** | Name of the station where the ride ended. |  |
| **end\_station\_id** | ride end station id |  |
| **start\_lat** | Latitude of the starting location. |  |
| **start\_lng** | Longitude of the starting location. |  |
| **end\_lat** | Latitude of the ending location. |  |
| **end\_lng** | Longitude of the ending location. |  |
| **member\_casual** | The membership type of a rider signifies if they are a regular user or a member. |  |

Table 4: Dataset features and description

Checked the missing values in dataset by using isnull().sum(), identified the missing value percentage for each variable as below, which is negligible and hence dropped the missing values.



The info function is used to identify the number of non-null values. The variables ‘started\_at’, ‘ended\_at’, ‘start\_station\_id’, ’end\_station\_id’ do not match the characteristics of feature and the data type. So converted the started\_at, ended\_at to date and time format, start\_station\_id,end\_station\_id to float type. Tried converting ‘start\_station\_id’ and ‘end\_station\_id’ from object to float type using function astype(). However, it resulted in error as there were values like SYS016, SYS025 etc in the column that could not be converted. Hence, pd.to\_numeric function is used with the errors='coerce' parameter, which converts any non-numeric values to NaN. This ensures that the conversion does not fail. These NaN values are added and hence dropped the observations containing NaN values. Checked for the duplicate rows. The unique function indicates that there are 2120 unique start station names and 1903 unique end station names, 2 unique ride type in the dataset. Verified if there are any invalid values in latitude and longitude observations.

* + 1. Feature engineering and transformation:

Added additional columns ‘start\_hour, ‘end\_hour’, ‘day\_of\_week’, ‘start\_date’,’ end\_date’, ‘day\_of\_week ’by extracting values from the ‘started\_at’ and ‘ended\_at’ variables that contains timestamp values by using function .dt.hour(), dt.day\_name(), dt.date() and .dt.dayofweek.

A new feature ‘ride\_duration’ is derived from existing data, for further analyse the number of bikes available at different station at specific hour, day of the week, weekend and weekdays. These features are added to provide additional insights and improve the predictive power of machine learning models.

station utilization is calculated by using formula:

‘ride\_duration’=(newyork\_bikes['ended\_at']-newyork\_bikes['started\_at']).dt.total\_seconds() / 60

* + 1. Box Plot to identify outliers and skewness:

Box plot and displot are plotted for variables, 'start\_lat', 'start\_lng', 'end\_lat', 'end\_lng', 'start\_hour', 'end\_hour', 'ride\_duration'. Skewness is measured for all the variables, the data indicates a distribution dominated by shorter rides with a few very long rides. The station IDs are almost symmetric, the latitudes and longitudes have modest to moderate positive skewness, the start and end hours have moderate negative skewness(~-0.565), and the ride duration has a very high positive skewness(23.862269).

Outliers are observed in all the features. Hence, considered using IQR method to eliminate outliers. Any values that are Q1-(1.5 x IQR) and Q3 + (1.5 x IQR) are categorized as possible outliers in order to determine the distribution's interquartile range (IQR). After removing outliers, the data is sorted by lower and upper bounds.

After the outlier treatment by IQR method, dataset is now more symmetrical and less affected by extreme results, particularly for ‘ride\_duration’. As a consequence, the dataset is more normally distributed, which can enhance the accuracy and performance of machine learning models that rely on normally distributed input features.

# **Data Visualization**

The data in ‘dublin\_bikes’ and ‘newyork\_bikes’ datasets is visualized separately in ‘Dublin\_bike\_Analysis.ipynb’ and ‘Newyork\_bike\_usage.ipynb’ respectively. Below is the analysis for both the cities during 1st June 2024 to 15th June 2024.

## **Analysis for Dublin Bikes:**

### **4.1.1 Bar Plot**

**To Visualize average number of bikes available for each station:**

Heuston Bridge (North), Heuston Bridge (Central), Heuston Bridge (Car Park), Parkgate Street and Talbot Street has the maximum number of bikes available, averaging around 25 bikes. The least number of bikes are available at Parnell Square North, Eccless Street, Grangegorman Lower (Central), averaging less than 5.

**To visualize the number of docks available in each station between 1st to 17th June 2024**

The bar graph shows the differences in the quantity of docks accessible at different Dublin bike stations. Pearse Street and Heuston Station East have the fewest docks, indicating either underutilization or a need for greater capacity.

King Street North, Newman House, and Wilton Park have the maximum dock availability, indicating high demand regions. Moderate dock numbers at stations like Eccles Street East and Mountjoy Square East help to balance supply and demand.

More docks may be required at high-demand stations to enhance service, and a bike redistribution should be considered to guarantee even availability at all stations.

### **4.1.2 Line Plot**

**To Visualize average number of bikes available for each day of the week:**

Calculated the average number of bikes available on each day of the week using groupby() and mean() function. Refer section 4.2 in the Dublin\_bike\_Analysis.ipynb file.

It is observed that average number of bikes available across Dublin city is more(12 to 13) on saturday and sunday. We see a gradual decrease in the number of bikes available at the station from Monday to Thursday which indicates, bikes usage is higher on these days and there is gradual increase in number of bikes available from Thursday to Sunday which means the less bikes are being used on these days.

**To visualize average number of bikes and docks available for each day of the week:**

This plot shows how are variables ‘num\_bikes\_available’ and ‘num\_docks\_available’ are related. We can observe that as the number of docks available increases from Monday to Thursday, number of bikes gradually decreases and number of bikes available increases from Thursday to Sunday and we have lesser docks available for charging at the station.

**To visualize number of bikes available based on hourly basis.**

The plot shows that minimum number of bikes are available during 16:00 hour of the day and 7:00 hour of the day, which indicates maximum bikes are used in the city during these peak hours. There is a moderate usage of bikes during 11:00 to 14:00 hour of the day. The minimum usage is observed during 19:00 hr to 4:00 hr of the day.

**To visualize the top 10 stations with minimum and maximum utilization by days of the week**

Her we observe that Parkgate Street has maximum bike usage on Saturday

The line plot, displays the average usage of the top 10 stations by day of the week, identifies clear trends in station usage. There is more demand during leisure times, as seen by the fact that most stations, including Custom House and South Dock Road, reach their peak on weekends, particularly on Sundays. On the other hand, there is a discernible decrease on weekdays such as Tuesday and Thursday, with stations such as Smithfield and Georges Quay exhibiting notable fluctuations. Conversely, stations such as Heuston Station (Car Park) continue to be consistently utilized, indicating a sustained level of demand. Due to vacations and other leisure activities, this pattern indicates that weekends are busier than weekdays, which may indicate commuter-focused usage. Higher the station utilisation, lesser is the bike utilisation.

### **4. 1.3 Stacked bar plot**

**To Identifying the instances of no bikes available during the day of the week for every station**

A stacked bar plot is created, where each bar represents a day of the week and is stacked with different colours representing different stations. The legend helps identify which part of each bar corresponds to which station.

As per the plot, Dublin bike stations Parnell Square North, Eccles Street East, and Mountjoy Square West, Fitzwilliam Square East, had a high percentage of unavailability, particularly on weekends. There are mild cases in Benson Street and St. Stephen's Green East, especially in the latter half of the week. There are consistently few cases at stations like Grand Canal Dock, College Green, and Dame Street. Sundays and Saturdays are the days with the largest percentage of no-bike incidents, while Tuesdays and midweek are the days with moderate peaks at Eccles Street East.

**Recommendations based on Analysis:**

Allocate bikes more efficiently on the weekends, add more space to stations with high demand, and modify the availability of bikes during peak hours.

### **4.1.4 Scatter Plot on a Geographical Dublin Map.**

Bike Availability at Dublin stations during different hours is visualized by scattermap. The hour is chosen as animation frame, the bubbles on the map show the average number of bikes available at the particular station. Each bubble points at specific bike station and the color gradient shows the variation in the number of cycles.

## **Analysis for New York bikes:**

### Line graph:

To visualize number of rides per day: The graph indicates a large jump in bike rides on June 1, 2024, and then daily variations in the range of 50,000 to 70,000 rides. Even while there is a decline on June 2, the number of rides steadily rises to 60,000 on the following days, suggesting steady bike utilization without significant fluctuations. These fluctuations may be caused by variables such as the day of the week, the weather, or local events; therefore, more research may be necessary to identify the causes causing these trends and adjust bike-sharing systems appropriately.

### Count Plot:

**To visualize number of rides per hour:** Highest number of rides are observed during 17:00 hr and 18:00 hr of the day. Least rides are observed at 4:00hr and 3:00hr of the day. A gradual increase in number of rides is observed from 10:00hr to 16:00 hr of the day. There is a spike in number of rides from 7:00 hr to 8:00hr of the day.

**To visualize number of rides by Day of the Week:** highest number of rides are observed on Wednesday and least rides are taken on Sunday.

### Bar plot:

**To visualize number of rides for top 10 Start and End Station:**

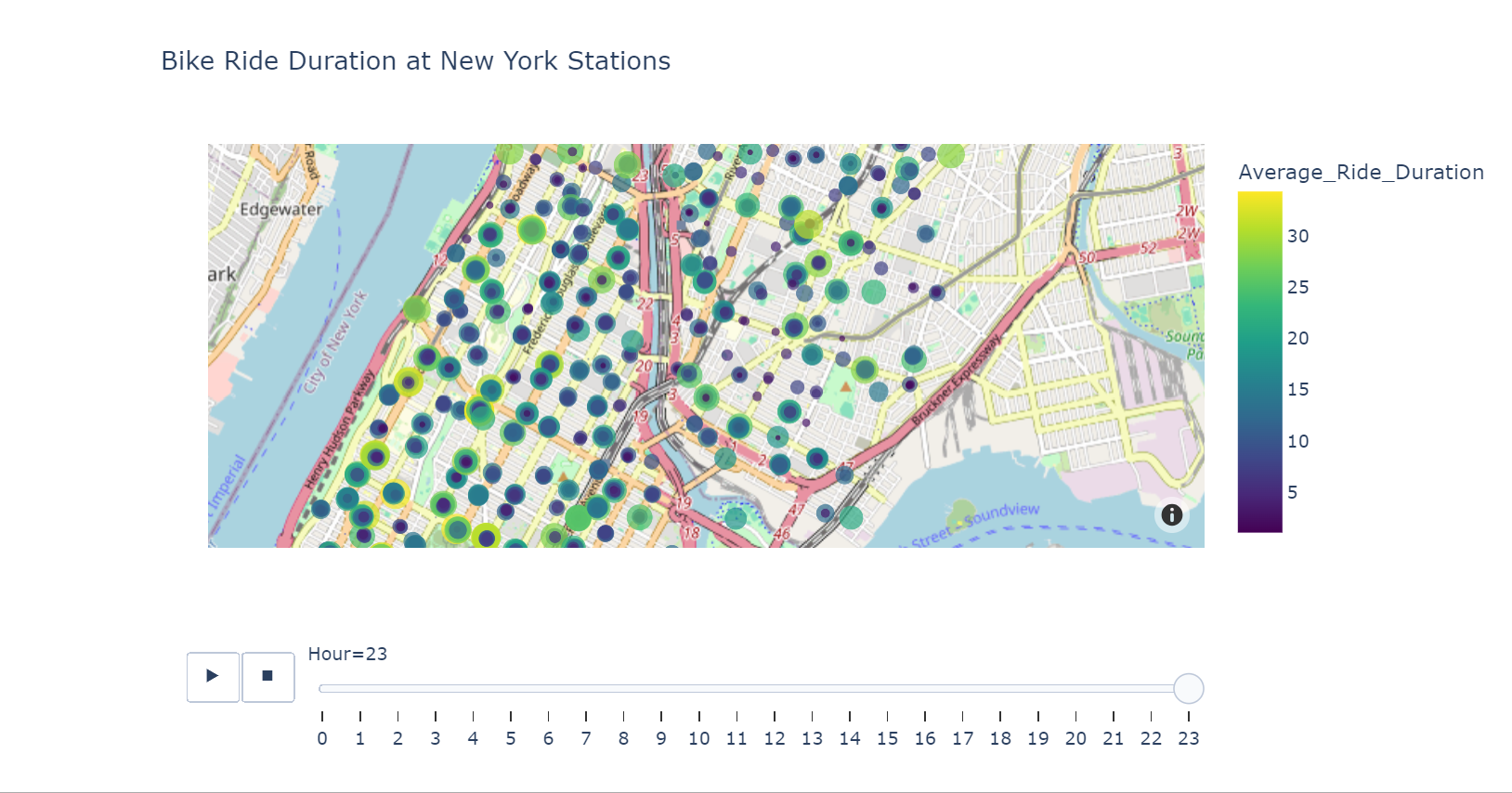
A thorough understanding of the bike-sharing trends can be gained by analyzing the top ten start and end points. First, 1 Ave & E 68 St., then West St & Chambers St., and University Pl & E 14 St. are the most often used start stations. Broadway & W 25 St., Pier 61 at Chelsea Piers, and W 21 St. & 6 Ave. are some more noteworthy start locations.

The most popular end stations are 12 Ave & W 40 St, followed by Lafayette St & E 8 St and E 33 St & 1 Ave. Important endpoints are also Cleveland Pl & Spring St, W 30 St & 10 Ave, and Broadway & W 58 St. The stations that appear on both top start and finish lists more than often, such as Broadway & W 58 St, indicate which particular routes riders utilize most frequently. The distribution and availability of bikes can be optimized with the use of this data.

### 4.2.4 Scatter plot on geographic New York map:

**To visualize the bike ride durations at New York stations at specific hour:**

The hour is chosen as the animation frame, the bubbles on the map show the station name, hour chosen, average ride duration at the specific location. The color gradient scale at the right indicates the variation in average ride duration at the specific location.



**To visualize Number of Rides at New York Stations Over Time:**

The data for number of rides at start station at specific hour is extracted by groupby function. The hour is chosen as the animation frame, the bubbles on the map show the station name, hour chosen, number of rides at the specific location. The color gradient scale at the right indicates the variation in number of rides at the specific location.

# **Dash for Visualization**

Dashboard is a collection of visualizations that present relevant information on a single screen, allowing users to easily access and explore detailed data. They provide a unified display to decision-makers, using visualizations such as maps, heatmaps, time-series, bar charts, histograms, radar charts, and pie charts. They are designed to provide an overview of the data and facilitate decision-making for non-expert users in visualization and data analytics. (sciencedirect, n.d.)

The data is plotted using scattermap and lineplot to visualize Dublin bike availability at different stations at an chosen date and hour of the day. The map shows visual representation of number of bikes available , the dropdown is designed to select the hour(00 to 24 hr), date can be selected from June 1st to June 17th in the animation frame.

For a selected hour in the dropdown and chosen date in the animation frame, map displays the number of bikes available at different chosen hr of the particular date at various Dublin stations. For the selected hour, line graph indicates the average number of bikes available from June 1st to June 17th at chosen hour in the dropdown.

Tufts Principle used for Dash: It focuses on using data visualization to effectively communicate information. (moodle.cct.ie, n.d.)

1. **Data-Ink Ratio:** Both visualizations efficiently utilize the data-ink ratio. The map maximizes the data-ink ratio by emphasizing data presentation above decorative components, efficiently representing data (bike availability) through color intensity. The chart uses a simple line graph to convey changes over time without unnecessary embellishments.
2. **Chartjunk:** The hover tooltips follow Tufte's minimalist approach by providing accurate data points without overcrowding the map. The lack of ornamental components that do not have a practical use in the chart is consistent with Tufte's principle for chartjunk. It emphasizes facts and illustrates patterns with a straightforward line.
3. **Graphical Integrity:** Color coding in map ensures that the data is appropriately shown, and tooltips give precise figures for individuals who are interested in particular specifics. Efficient data interpretation is ensured by the well-labeled axes and the tooltip that provides exact data values in line graph.
4. **Data Density:** A daily summary of bike availability is provided by the graphic, which is useful for identifying trends over time.
5. **Appearance and Usability:** Users can understand the entire scenario by looking at the map, which effectively conveys a lot of data spatially. The interactive features, such as date/time selections and hover tooltips, offer an efficient design while allowing dynamic data exploration for users. The chosen colors and layout improve the visual appeal and usefulness of the design.

The line chart successfully represents data points with a clean and smooth line. User experience is improved with interactive tooltips, which let them examine data points in detail.

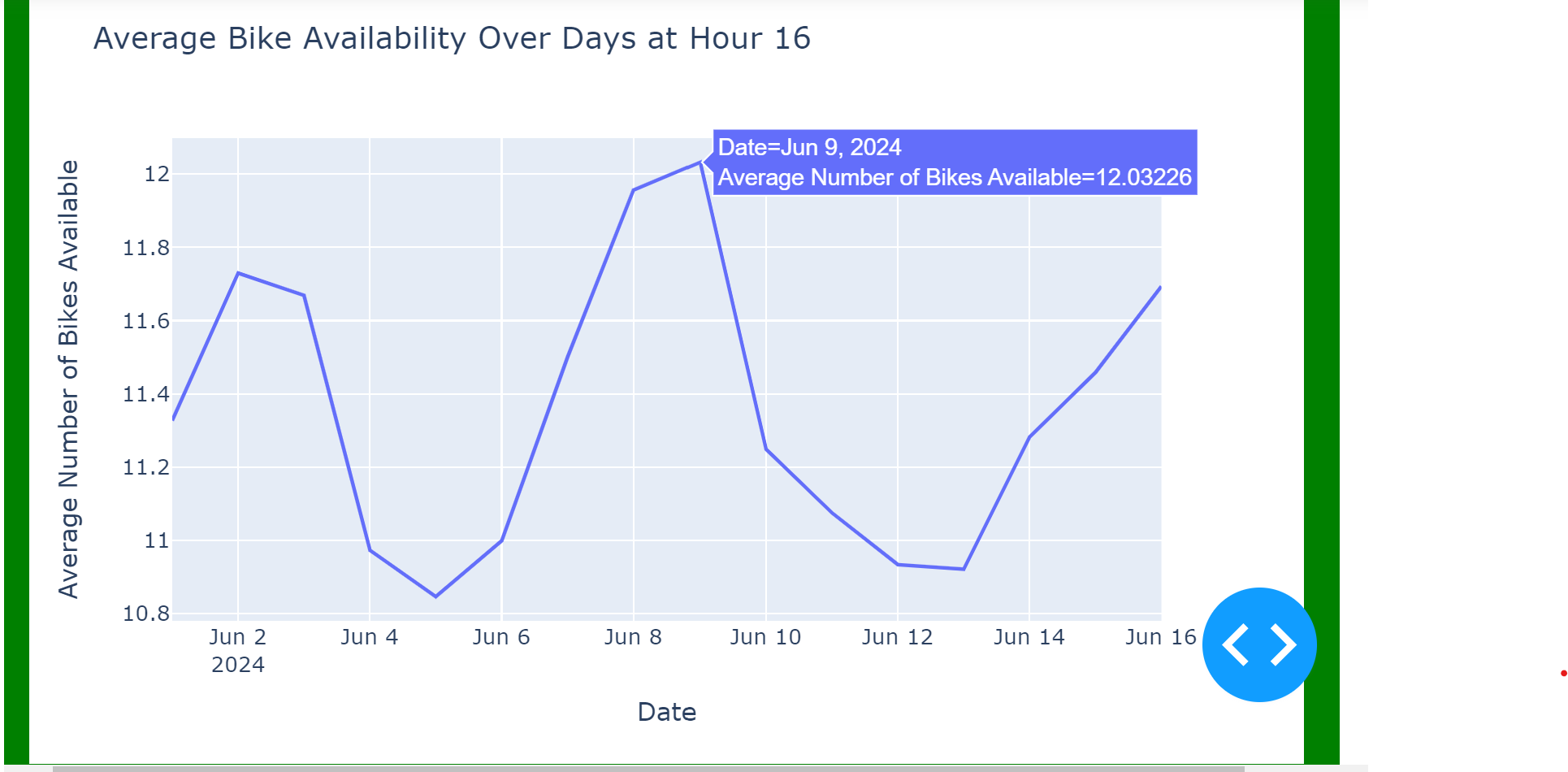
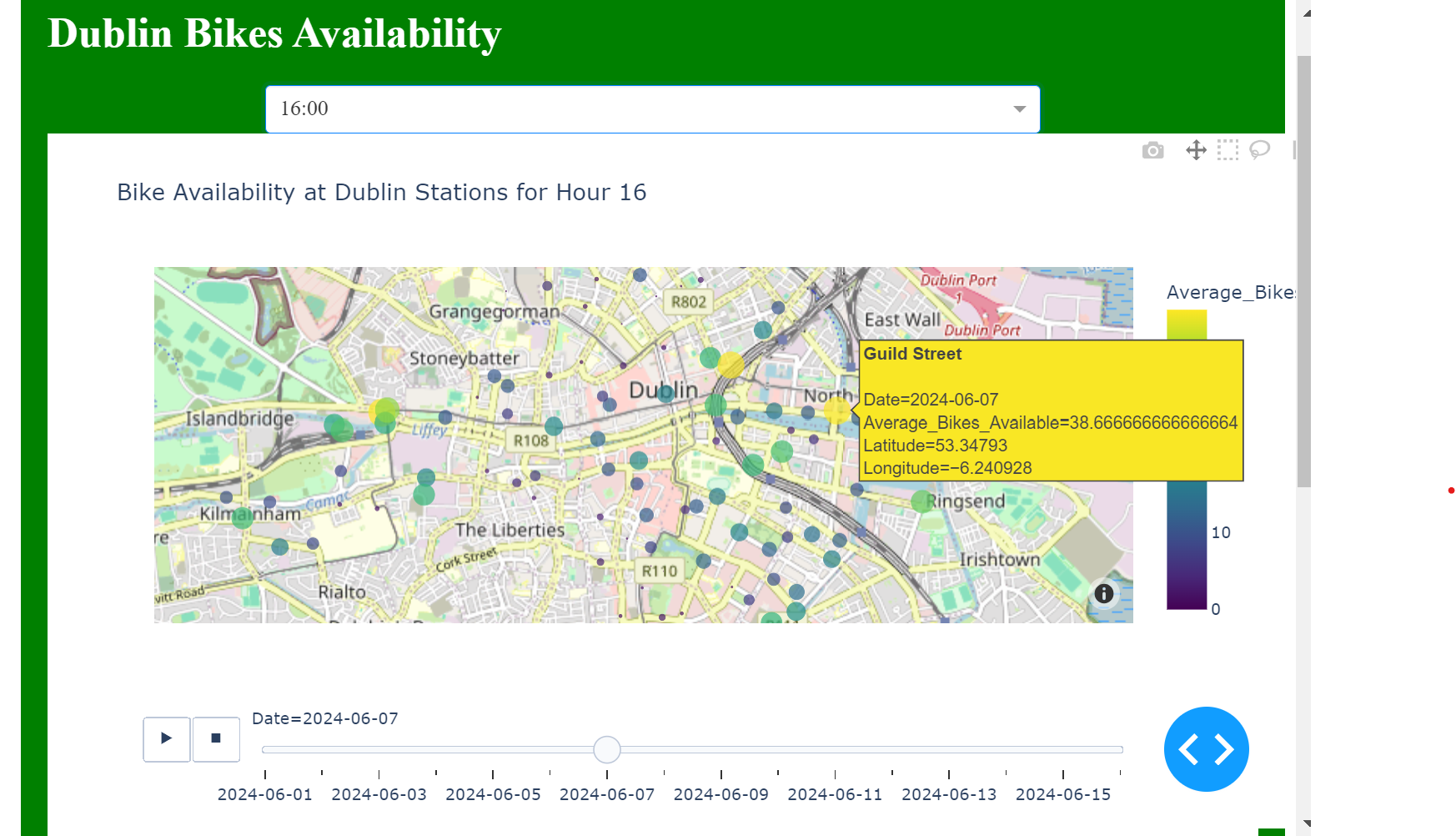


Figure 1 Dash to visualize number of bikes available in Dublin

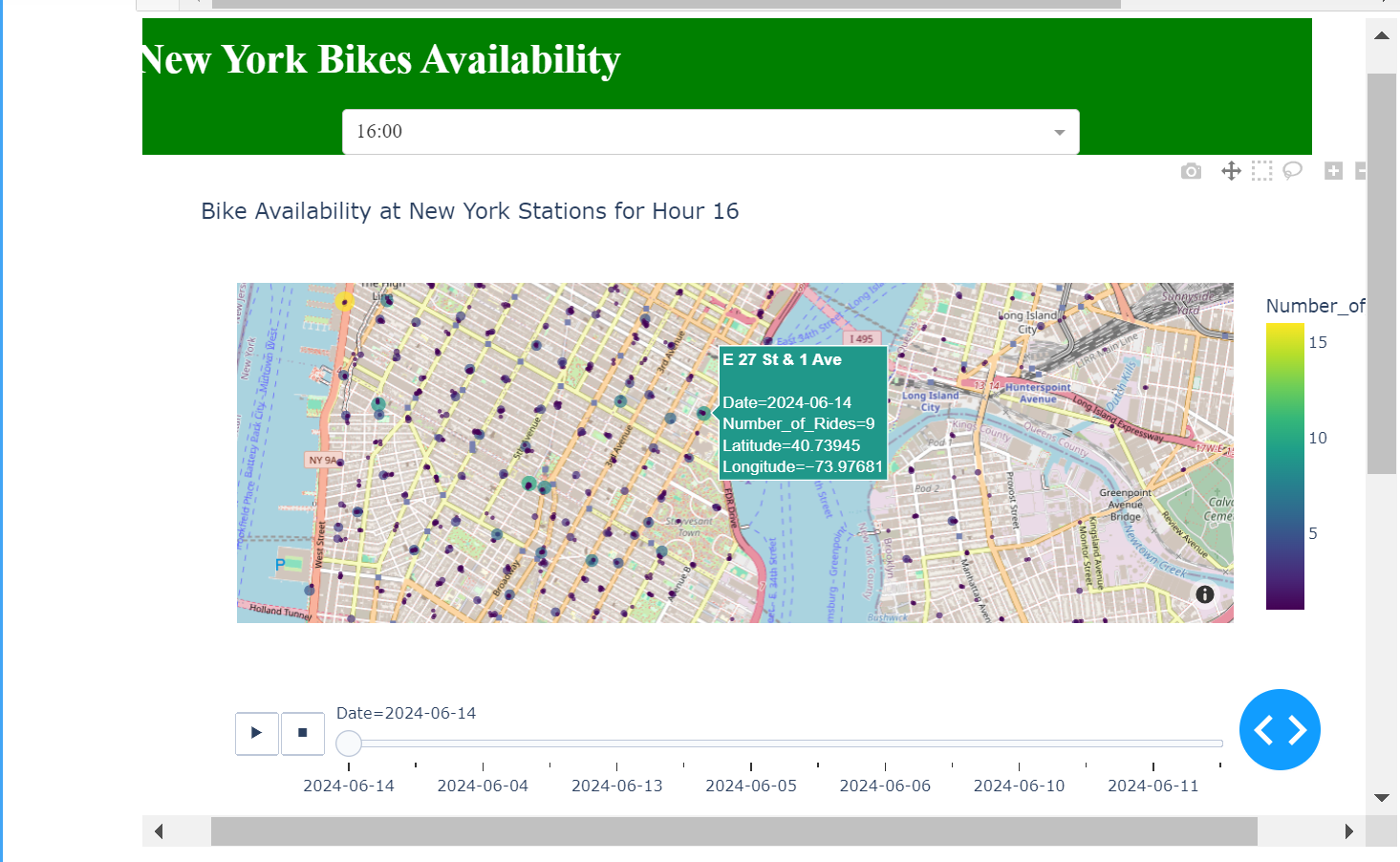




Figure 2: Dash to visualize NewYork Bikes availability.

As per the data visualization performed for Dublin and New York city bikes, It is observed that New York has a higher utilization and better infrastructure for cyclists. The number of stations available are significantly higher in NewYork (2103) when compared to Dublin(114). This aspect could be due to the geographical area which is only 115 km² for Dublin and 1,213 km² for NewYork and the difference in population between these cities.

# **Data Pre-processing for Dublin\_bikes.**

## **Feature Extraction:**

The features ‘hour’, ‘day\_of\_week’ is extracted from the variable ‘time\_stamp’ using the Datetime Properties. dt.hour and .dt.dayofweek() from pandas package.

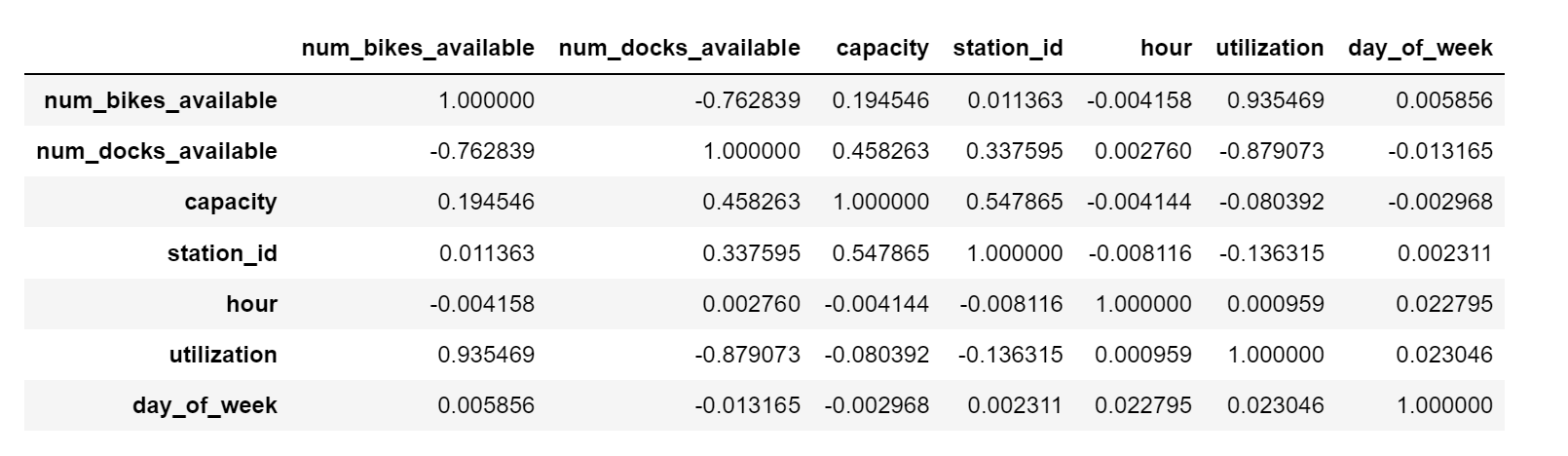
The feature ‘utilization’ is derived from the existing data ‘**num\_bikes\_available’** and ‘**capacity**’ of the station.

Station utilization is calculated by using formula:

**‘utilization’ = [(‘num\_bikes\_available’)/ (‘capacity’) ]\* 100**

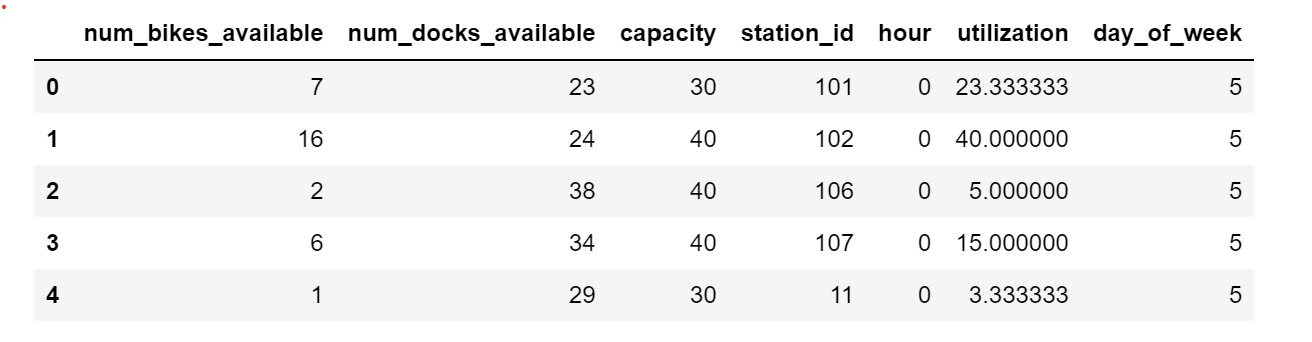
## **6.2 Feature Selection:**

The function corr() is used to identify the correlation between the variables 'num\_bikes\_available', 'num\_docks\_available', 'capacity','station\_id','hour','utilization','day\_of\_week'. The heatmap is plotted to identify the co-relation between the features. It is observed that number of bikes available and utilization of station are highly co-related. Hence, I am choosing to predict the number of bikes available based on features that are co-related to the target variable 'num\_bikes\_available'.

*Figure 3: Correlation between variables*

Plotting "num\_bikes\_available" and "utilization" using regplot() revealed a linear relationship between them because they are significantly co-related. The data distribution shows a linear regression model fit.

Selected the features by using user defined function getCorrelatedFeature () setting the threshold value as 0.0.

Figure 4: Selected features for Machine learning model – dataset ‘dublin\_bikes\_corr’

**Note:** Encoding is not performed as all the required data for machine learning model is in numerical form.

## **Scaling:**

Feature scaling is a vital pre-processing step in machine learning that involves transforming numerical features to a common scale. Scaling techniques aim to normalize the range, distribution, and magnitude of features, reducing potential biases and inconsistencies that may arise from variations in their values. (medium, n.d.)

Standard scaling is performed to transform the features to have a mean of 0 and a standard deviation of 1. It subtracts the mean of the feature and divides it by the standard deviation.

**z = (x - u) / s,**

where u is the mean of the feature values,

s is the standard deviation of the feature values.

Standardization is used as we have seen the data distribution plot and skewness values for most of the features are close to zero, indicating Gaussian distribution, less variation in the data points for each feature.

# **Machine Learning**

Supervised machine learning algorithms are used for predictive analysis where the model is trained on a labelled and scaled dataset ‘dublin\_bikes\_corr\_scale’. Data pre-processing is performed by feature selection (Refer section 6.2), here the number of bikes available in Dublin at various stations ‘**num\_bikes\_available’** is considered as the dependent variable or target variable ‘y’ and other features are considered as predictors or independent variable ‘X’.

As it is analysed that ‘**num\_bikes\_available’** is highly correlated **with ‘utilization**’ (Refer Figure 2), plotted the regplot() to find the linear relationship between the features and identified that data fits linear regression model.

The target variable and independent variables used for all the ML models are same. Data is split into testing and training, for Linear regression models training data considered is 70% and testing data is 30%(0.3). For Decision tree, Random Forest the training and testing data is considered as 80% and 20% respectively.

## **7.1 Linear Regression:**

Linear regression is an algorithm that provides a linear relationship between an independent variable and a dependent variable to predict the outcome of future events. (spiceworks, n.d.)

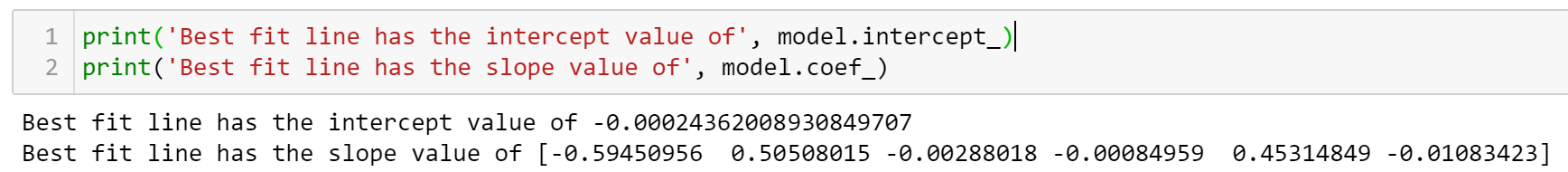
Mathematically, **y=β0​+β1​x1​+β2​x2​+β3​x3​+β4​x4​+β5​x5​+β6​x6,**

where, y is the predicted value of ‘num\_bikes\_available’.

β0​ is the intercept of the line.

β1,β2,β3,β4,β5,β6 are the slopes corresponding to each independent variable x1,x2,x3,x4,x5,x6.

Evaluated the intercept and slope values of the best fit line to understand the relationship between dependent and independent features.



The value indicates, bike availability is significantly impacted negatively by the number of docks available, according to the linear regression model, with each additional dock resulting in a reduction of roughly 0.595 in the total number of bikes. On the other hand, bike availability is positively impacted by station capacity and usage, increasing it by 0.505 and 0.453, respectively, for every unit increase. The station ID, day of the week, and hour of the day have negligible effects, suggesting that they are not as important in predicting the availability of bikes. When all predictors are 0, the intercept is almost zero, indicating a negligible baseline value.

Performed analysis using other models like Decision tree, Random Forest and Gradient boosting and below is the comparative analysis observed.

## **7.2 Comparative Analysis of different ML models used:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **R² Score** | **Mean Absolute Error(MAE)** | **Mean Squared Error** | **Root Mean Squared Error(MSE)** |
| **Linear Regression** | 0.9766 | 0.086 | 0.0234 | 0.153 |
| **Decision Tree** | 0.9999998091 | 2.0096e-06 | 1.9128e-07 | 0.0004 |
| **Random Forest** | 0.9999999228 | 2.6829e-06 | 7.7344e-08 | 0.0003 |
| **Gradient Boosting** | 0.9996 | 0.0157 | 0.0004 | 0.0212 |

Table 5: Comparison of different ML models

**Linear Regression:** With an R2 value of 0.9766, linear regression offers an excellent fit. However, in comparison to other models, its errors are larger (MAE: 0.0860, RMSE: 0.1530).

**Decision Tree:** With an R2 value of 0.9999998091 and incredibly low errors (MAE: 2.0096e-06, RMSE: 0.0004), the decision tree achieves almost perfect accuracy.

**Random Forest:** The most accurate model is Random Forest, which outperforms all other models with the best R² score of 0.9999999228 and the lowest errors (MAE: 2.6829e-06, RMSE: 0.0003).

**Gradient Boosting:** With an R2 score of 0.9996 and minimal errors (MAE: 0.0157, RMSE: 0.0212), gradient boosting performs well, however with a little less accuracy than tree-based models.

**Conclusion:** The best option is Random Forest, closely followed by Decision Tree. While Linear Regression is more appropriate for simpler datasets where interpretability is crucial, Gradient Boosting provides great performance with slightly larger errors.

## **7.3 Hyperparameter Tuning:**

Hyperparameters are configuration variables that are set before the training process of a model begins. They control the learning process itself, rather than being learned from the data. Hyperparameters are used to tune the performance of a model, and they can have a significant impact on the model’s accuracy, generalization, and other metrics. (geeksforgeeks, n.d.)

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **R² Score** | **MAE** | **MSE** |
| **Decision Tree (Tuned)** | 1 | 2.01E-06 | 1.91E-07 |
| **Ridge Regression** | 0.9766 | 0.086 | 0.0234 |
| **Lasso Regression** | 0.9758 | 0.0932 | 0.0243 |

Table 6: Comparison of different ML models after hyperparameter tuning

The hyperparameter tuning is performed for the Decision Tree Regressor, Ridge Regression, and Lasso Regression models using GridSearchCV.

For Decision tree, max\_depth values [None, 10, 20, 30] are selected, minimum number of samples required to split node as [2, 10, 20], minimum number of samples required to be a leaf node as [1, 5, 10].All these values are processed for combinations of the specified hyperparameter values using 5-fold cross-validation for the model.

For Ridge and Lasso Regression, alpha values [0.01, 0.1, 1, 10, 100] and [0.01, 0.1, 1, 10, 100] are chosen respectively, to perform 3-fold cross validation. Using Grid Search, cross validation and hyperparameter values, the best hyperparameter values are selected that produce best R² score for the machine learning model.

**Conclusion:**

Ridge Regression's performance remains constant with Linear Regression, suggesting no significant rise or change in metrics after adjustment.

Lasso Regression shows a slight reduction in R² score and an increase in MAE and MSE compared to Ridge Regression, indicating that it may not perform as well as Ridge Regression after tweaking.

## **7.4 Generalization using Cross validation of ML models:**

Cross-validation is a statistical method of evaluating generalization performance that is more stable and thorough than using a split into a training and a test set. (Sarah) (Guido)

The R² score is utilized as the evaluation metric when doing 10-fold cross-validation on the models using cross\_val\_score. X (all features in ‘dublin\_bikes\_corr\_scale’) and Y (‘num\_bikes\_available’) are the data that are utilized.

|  |  |  |
| --- | --- | --- |
| **Model** | **R² Score (Cross-Validation)** | **Description** |
| **Linear Regression** | 0.9699 | About 96.99% of variance is explained by the model. |
| **Gradient Boosting** | 0.999505 | 99.95% of the variance in the data is explained by the model, indicating excellent performance. |
| **Decision Tree** | 0.999997 | About 99.99% of variance is explained by the model, indicating high accuracy. |
| **Random Forest** | 0.999996 | About 96.99% of variance is explained by the model. |

Table 7: Cross validation for generalization of different models used in analysis

Higher scores denote better performance, and the R2 score shows how well each model matches the data. In comparison to the Linear Regression model, the Gradient Boost, Decision Tree, and Random Forest models have extremely high R2 values, indicating that they fit the data very well.

# **Machine Learning for newyork\_bikes dataset (Newyork\_bike\_usage.ipynb)**

The data pre-processing is performed similar to dublin\_bikes dataset. The features extraction is done to extract features like ‘**ride\_duration**’ (refer section 3.2.1 of this report).

Here, we are trying to predict the possible length of bike ride (ride duration) based on start station id, end station id, start latitude, start longitude, end latitude and end longitude, start hour and end hour.

Target variable considered is y = ‘**ride\_duration**’,

Other features are considered as the independent variables and are used to make predictions.

Feature selection process is followed to reduce the dimension of original dataset, by choosing the features that are highly correlated to ‘ride\_duration’**,** thusincluding only the required features in the dataset ‘newyork\_bikes\_corr’. The dimension is reduced to 9 features.

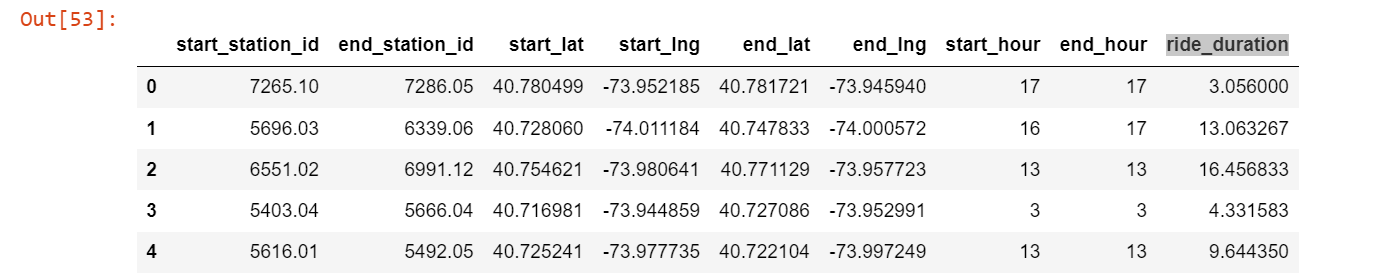


Table 8: The dataset used for ML models ‘newyork\_bikes\_corr’’

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **R² Score** | **Mean Absolute Error (MAE)** | **Mean Squared Error (MSE)** | **Root Mean Squared Error (RMSE)** |
| **Linear Regression** | 0.0059 | 0.801 | 0.9952 | 0.9976 |
| **Decision Tree** | 0.3949 | 0.4927 | 0.6058 | 0.7783 |
| **Random Forest** | **0.6684** | **0.3845** | **0.3319** | **0.5761** |
| **Gradient Boosting** | 0.3297 | 0.6417 | 0.6711 | 0.8192 |

Table 9: Comparative Analysis of Machine learning models

**R² Score** :

* Random Forest is the most likely to best capture the link between input features and ride time and to explain the greatest variance in the data, with an R2 score of 0.6684.
* Decision Tree has a reasonable fit, as indicated by its R2 score of 0.3949.   
  Gradient Boosting captures some variance but not as well as Random Forest or Decision Tree, as evidenced by its somewhat lower R2 score of 0.3297.
* Linear regression performs badly in explaining variance, as evidenced by its extremely low R2 score of 0.0059, which raises the possibility that it is inappropriate for this dataset.

**Mean Absolute Error (MAE):**

* With the lowest average prediction error (MAE) of 0.3845, Random Forest outperforms the others once more.
* Decision Tree outperforms Random Forest with a little higher MAE of 0.4927.
* Gradient Boosting is less accurate overall than the other two models, with an MAE of 0.6417.
* Compared to the other models, Linear Regression consistently generates greater errors, as evidenced by its highest MAE of 0.8010.

**Mean Squared Error (MSE):**

* With the lowest MSE of 0.3319, Random Forest has the lowest overall squared errors. This puts it in first place.
* An MSE of 0.6058 is found for Decision Tree.
  + With a little higher MSE of 0.6711 than the Decision Tree, Gradient Boosting is the preferred method.
* With the highest MSE of 0.9952, Linear Regression is the least robust to huge errors.

**Conclusion:** Based on all four criteria (RMSE, R2 Score, MAE, and MSE), Random Forest is the model that performs the best. It performs well in forecasting the length of bike rides since it has the lowest error metrics and the greatest R2 score.

## Dash to visualize machine learning model results.

A dashboard is made to show the results of the various machine learning algorithms that are applied in this study. (plotly.com, n.d.) To view the training, test set, and prediction that displays the actual vs. predicted values, select the model using the dropdown.



Figure 5: Dash for machine learning models

The fact that most of the blue dots are above the red line suggests that the ride duration is frequently overestimated by the model.

Variability in prediction accuracy is indicated by the dispersion of blue dots surrounding the red line, particularly as the true ride duration grows.

# **Programming for Data Analytics**

**Data from Diverse source:**

The raw data for analysis is downloaded from <https://data.gov.ie/dataset/dublinbikes-api>. And <https://data.gov.ie/dataset/dublinbikes-api>. The data for Dublin bikes is downloaded in csv format, whereas the data for NewYork is downloaded in .zip format and ‘202406-citibike-tripdata\_1.csv’ is used for our analysis.

**Data Manipulation:**

Using the library pandas.series.dt() package, more features are extracted from raw data. It is used to extract dt.hour, .dt.dayofweek(), dt.date, dt.day\_name(), dt.total\_seconds() data from timestamp values provided in original dataset. (Refer Newyork\_bike\_usage.ipynb line #24).

Properties like to\_datetime are used to convert data type of features indicating date and time.

Different ports are used to run multiple dash, as I have used two jupyter notebooks, port 8052,8051 are used to view the data at the same time.

**Data structures**: Both the datasets created from raw data are downloaded in .csv format.

**Testing:**

Software testing is a way of validating whether a software is performing as a customer or product consumer would expect. (medium.com, n.d.) Testing prevents errors, bugs and defects in the code and helps deliver a quality software product.

In the jupyter notebook Dublin\_bike\_Analysis.ipynb, Line #10, the dataset dublin\_bikes has a shape (315747, 15). I have dropped the column "short\_name","region\_id" as there are zero observations for these variables. There is a validation performed to check the columns are dropped in the next step by **.shape** instruction.

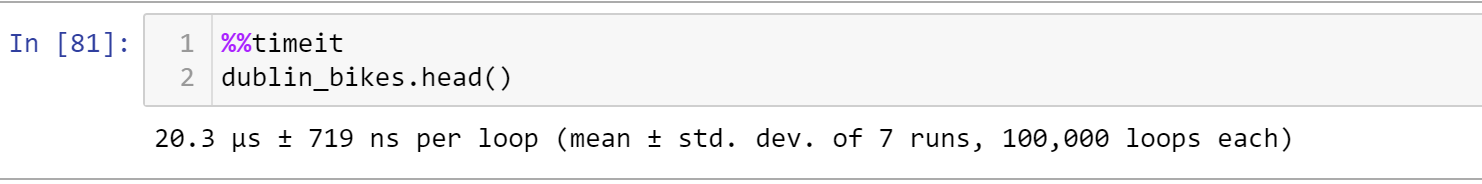
In Newyork\_bike\_usage.ipynb, line #36, IQR method is used to remove outliers by using a user defined function remove\_outliers\_iqr(), after the outliers are removed, box plot is re-plotted to observe outliers are removed and skewness is also verified to make sure, the function remove\_outliers\_iqr() has performed its intended functionality.

**Optimisation:**

At line #15, performed a check to identify any duplicate rows using function duplicated. The output is stored in duplicate\_rows\_df. However, in the next step, the dataset is deleted to free the memory space.

In section 4.4 Visualizing num\_bikes\_available based on hour, the line plot for multiple features ‘avg\_bikes\_available’, ‘median\_bikes\_available’ and ‘std\_bikes\_available’ is drawn by storing the values in array and using ‘for loop’, avoiding repetitive code instructions.

The dataset ‘dublin\_bikes\_corr’ is created by identifying the correlation between features. This feature selection process reduces the dataset dimension by focusing only on required data for machine learning model. The dataset features are reduced from 15 to 7. This in turn improves training speed and also helps in improving model accuracy. This also reduces the code execution time as seen below. (www.linkedin.com/pulse/feature-engineering, n.d.)



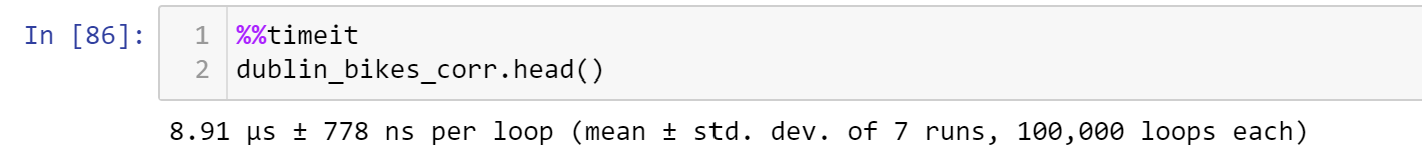


Figure size is allocated to all the plots. Figure size determines how much screen area each plot should occupy, which can help control memory usage and rendering performance. When working with a large number of plots or high-resolution displays.

Optimization by List: List is created for columns that ought to be displayed in the plot. This prevents pointless calculations on erroneous information. Because it avoids the overhead of verifying the data types for every column and concentrates just on pertinent data for visualization.

# Reference

<https://plotly.com/python/choropleth-maps/>

<https://plotly.com/python/ml-regression/#ml-regression-in-dash>

<https://dash.plotly.com/layout>

(Andreas C. Müller and Sarah Guido, no date)

Introduction to Machine Learning with Python

(geeksforgeeks, n.d.)

[https://www.geeksforgeeks.org](https://www.geeksforgeeks.org/functional-programming-paradigm/)

(feature-scaling-machine-learning-normalization-standardization)

<https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/>

NumPy, 2023. *About us.* [Online]   
Available at: <https://numpy.org/about/>

Matplotlib, T., 2023. *Matplotlib: Visualization with Python.* [Online]   
Available at: <https://matplotlib.org/>

Waskom., M., 2023. *seaborn.* [Online]   
Available at: <https://seaborn.pydata.org/>

developers, s.-l., 2023. *index.* [Online]   
Available at: <https://scikit-learn.org/stable/index.html>

community, T. S., 2023. *Statistical functions.* [Online]   
Available at: <https://docs.scipy.org/doc/scipy/reference/stats.html>

developers, s.-l., 2023. *1.11. Ensembles: Gradient boosting, random forests, bagging, voting, stacking.* [Online]   
Available at: <https://scikit-learn.org/stable/modules/ensemble.html>

Foundation, P. S., 2001-2023. *warning.* [Online]   
Available at: https://docs.python.org/3/library/warnings.html

pydata, p., 2023. *pandas.* [Online]   
Available at: <https://pandas.pydata.org/>

Waskom, M., 2023. *seaborn.heatmap.* [Online]   
Available at: <https://seaborn.pydata.org/generated/seaborn.heatmap.html>

Ptil, P., 2018. *Medium/ What is Exploratory Data Analysis?.* [Online]   
Available at: <https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15>

Pandas, 2023. *pandas.DataFrame.* [Online]   
Available at: <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.head.html#pandas.DataFrame>

Simplilearn, n.d. *The Complete Guide to Skewness and Kurtosis.* [Online]   
Available at: <https://www.simplilearn.com/tutorials/statistics-tutorial/skewness-and-kurtosis>

Sklearn, n.d. *sklearn.linear\_model.LinearRegression¶.* [Online]   
Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html>

<https://www.linkedin.com/pulse/feature-engineering-its-types-uttam-kumar>

<https://medium.com/@shivanipickl/what-is-feature-scaling-and-why-does-machine-learning-need-it-104eedebb1c9#:~:text=Feature%20scaling%20is%20a%20vital,efficient%20model%20training%20and%20performance>