## Table of Content

[**Introduction**](#_96x78dy531aw) **2**

[Part 2 - Programming](#_ptm87yfbgewa) 2

[Data Preparation](#_t7w9au2enjdz) 2

[Data Structure](#_8pz841uomzpg) 3

[Missing Values](#_lewt6pbubbey) 3

[Statistical Analysis](#_ol39ta216rvn) 3

[Visualizing Data Distributions](#_zci1qprhle9e) 3

[Identifying Outliers](#_46yss6pd2wbu) 4

[Examining Relationships](#_4x80ijpyuctm) 4

[Data Cleaning and Engineering](#_h89ipgyacepk) 4

[Handling Missing Values](#_10eobgsffn23) 4

[Removing Duplicates](#_k0m3j758zwe) 4

[Converting Data Types](#_3zpl4rnuwhc3) 5

[Feature Engineering](#_6eygsp8fxuyo) 5

[Machine Learning Models](#_qyr4cgnk54b7) 5

[Clustering with KMeans](#_slqnqtro06w6) 5

[Preprocessing Function](#_8wjx3tng1qi4) 5

[Part 2 - Data From Diverse Sources](#_gdor8i7xsb0f) 6

[Data Acquisition](#_65y6oixjm8qg) 6

[Dublinbikes Dataset](#_suhvk2z7g1qe) 6

[Citi Bike Dataset](#_cgoltnihqsdk) 6

[Data Integration](#_2l3a0770i06u) 6

[Comparing Data Structures](#_i68drmb2ikn1) 6

[Processing Data from Diverse Sources](#_bvz6pmw8d63e) 7

[Aggregation Methods](#_8cw7wpebmmr0) 7

[Part 3 - Data Manipulation](#_bnngfw2zpvan) 7

[Dublinbikes Dataset](#_f8sg36xi3j7y) 7

[Citi Bike Dataset](#_hhnjykgtdwzb) 7

[Part 4 - Data Structures](#_6karotvtjmca) 8

[Data Formats](#_ame9z4tgjg5y) 8

[CSV Format](#_ykvaepusmf9g) 8

[JSON Format](#_z70gy1yjtqln) 8

[Processing Data in CSV Format](#_inv5hurpoqyz) 8

[Dublinbikes Dataset](#_lpsb7x42kuid) 8

[Processing Data in JSON Format](#_w2dad87t10e7) 9

[Citi Bike Dataset](#_fqvttqvjo6md) 9

[Comparison of Data Processing Techniques](#_1bkqhp4kgsop) 9

[CSV Processing](#_j8h41fqezxqc) 9

[JSON Processing](#_u6tuhogrtscr) 9

[Part 5 - Testing Strategy](#_f38glrhxdvxu) 10

[Testing Strategy](#_p0rrm53vvfj2) 10

[Unit Testing](#_z69cl9ovro01) 10

[Integration Testing](#_s4rxqu1smgt4) 10

[Trade-offs in Testing](#_yasf3iwytqiq) 10

[Part 6 - Optimisation](#_ln12m8tshzll) 11

[Optimization Techniques](#_m71s6f97oluk) 11

[1. Efficient Data Loading](#_g398zi9l0ig7) 11

[2. Data Preprocessing Optimization](#_vbw5hjz7mvs1) 11

[3. Memory Management](#_p5sf5qwn2uqr) 11

[4. Parallel Processing](#_cppdavpw6s9p) 12

[5. Efficient Data Storage](#_h8ll2ekfjddn) 12

[Trade-offs in Optimization](#_fyj7zr5mejx8) 12

[Conclusion](#_1yyo48mtzl3g) 12

## 

## Introduction

This paper will focus on the exploration of the Dublinbikes and Citi Bike datasets with regards to availability of bikes, usage of stations, and geographical locations. This dataset was obtained from Dublin City Council and it focuses on bike stations in Dublin; number of bike stands, available bikes and geographical coordinates infromation is presented. Likewise, Citi Bike data obtained from the NYC Open Data source offers similar data of bike-sharing stations in New York City. The objective of this comparative study is to increase the knowledge on bike usage in the two cities and possible recommendations to make for improvement in the bike-sharing systems.

To make the analysis fully automated, Python tools and libraries are used while implementing the analysis in a Jupyter Notebook with high attention to the code quality. Knowledge of the riding profiles of the Dublinbikes as well as the Citi Bike systems is therefore important towards enhancing performance of the cycling systems, satisfying the users as well as encouraging eco-friendly transport. The report follows a structured approach: They include data pre-processing, EDA, data cleaning, feature engineering, applying of Machine Learning algorithms, and data interpretation and visualization. Every part delivers clear descriptions and analyses of the applied methods and results, thus, offering the required level of clarity upon the process and results of analysis. This structure of sections allows for the effective analysis and comparison of bike sharing systems in Dublin and New York City for network improvement and enhanced sustainable mobility.

### **Part 2 - Programming**

#### **Data Preparation**

Data cleaning is one of the most critical and must-do activities from every data analysis business intelligence project. The Dublinbikes data set was obtained from the Dublin City Council open data site; Dublinbikes was made available under an open data license which means that anyone can use it, alter it, and distribute it so long as the owner of the data is credited. Regarding the data type format, the dataset is provided in the CSV format which is versatile and can be useful for analysis.

**Exploratory Data Analysis (EDA)**

The technique of visually describing structures and characteristics of the data is called Exploratory Data Analysis (EDA). In the case of EDA, it does assist in potential points of interest for further examination, or areas that seem to deviate from expectation. It includes descriptive analysis in terms of the generation of summary statistics, the visualization of distributions and relations between the variables.

##### **Data Structure**

Some structure of the dataset is also analyzed to determine the type of data and how they are organized. This includes the extraction of the name of columns in the datasets, data type of values in the columns as well as the existence of any missing values in the dataset. Knowledge of the structures of the data is important as it aids in planning for the next stages that involves cleaning the data and analyzing it. It is from this view that one is able to notice any initial tendencies or defects in the data that might warrant their correction; for instance, missing entries or squashed data types implicitly in the parameter.

##### **Missing Values**

One of the processes of EDA is to identify and also display the missing values in the data set. Certain kinds of values may be missing and this can present a problem which must be managed correctly. To elaborate, heat maps offer a visual way to determine the exact location of the missing data and realize its precise scope, which subcolumns need to be tackled during data cleaning sessions. This step assists in determining the strategy for dealing with the missing values, whether to impute, remove, or use other methods.

##### **Statistical Analysis**

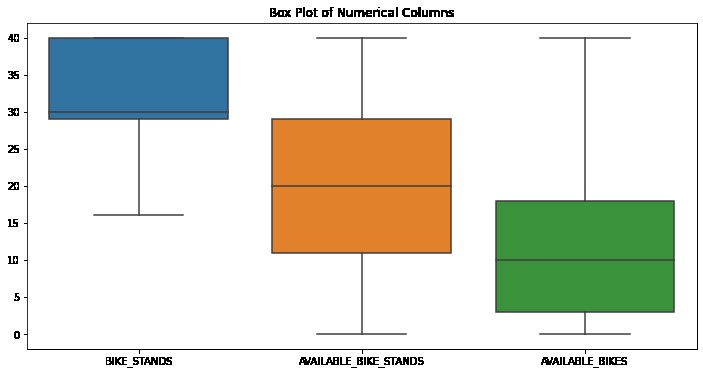
Computation of summary statistics of numerical columns assist in figuring out the distribution and central measures. Descriptive statistics gives the measures of central tendencies such as mean, median and measures of dispersion such as standard deviation and variance as well as giving the general distribution form of the numerical variables in the data set. This step is critical in the process of identifying any possible problems regarding the quality of data, including the problems like outliers or distorted data and it also gives the basic understanding of the dataset that is going to be used in the next steps.

##### **Visualizing Data Distributions**

Histograms are employed for the analysis of the distribution of some of the important variables for instance, number of bike available. This assists in giving the frequency distribution of the number of bikes in the various stations in order to better serve the customers. The histogram of available bikes depicts the station distribution of fully or partially sold out bicycle stations and gives an idea of how frequently it occurs. To do this, there exists several distributions which when understood will help in finding out patterns of the data and outliers.

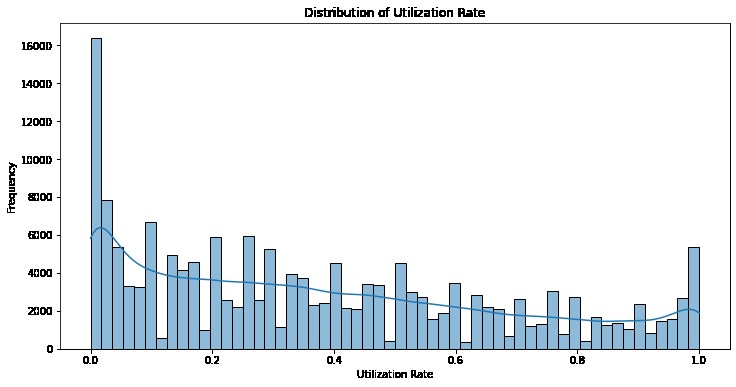
##### **Identifying Outliers**

One can use box plots to identify extreme values in the numerical columns. They can be defined as the values that are quite different from the remaining values and can influence the results. Outliers are critical to analyze as they allow identifying the range of possible values and make the first decisions regarding data cleaning and data preprocessing. This string represents exceptions that may imply non-standard behavior or mistakes that have to be dealt with when interpreting the obtained results. Below is the box plot and our data don’t have outliers.



##### **Examining Relationships**

Scatter plots are applied best for studying the connection between certain attributes, for instance, bike stands and available bikes. Knowledge of these relationships is essential for distilling the system’s characteristics into patterns and interdependencies. For instance, where the number of the bike stands is proportional to the number of available bikes, this implies that most passenger stations with many bike stands will have many bikes to support that capacity. This enables the assessment of how different variables are related in the implementation of the research question.

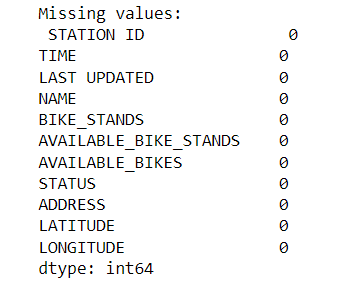


#### **Data Cleaning and Engineering**

It comprises missing values in a dataset, removal of a duplicate record, and variables converting in a data set. The above steps are important when it comes to the validation of data and maintaining data integrity. Feature engineering can be defined as the process of deriving new features or transforming existing ones in order to improve the analysis performed on the dataset.

##### **Handling Missing Values**

The process of managing missing values is considered to be one of the most important subprocesses in the data cleaning phase. These missing values can even pull the results and subsequently the conclusions, in a very wrong direction. The techniques like forward fill make sure that the data doesn’t get lost in the process and the dataset remains intact. Popular for its ability to keep the order of data intact, this method replaces the missing values with the previous non-missing value. Below is shown all column missing value count



##### **Removing Duplicates**

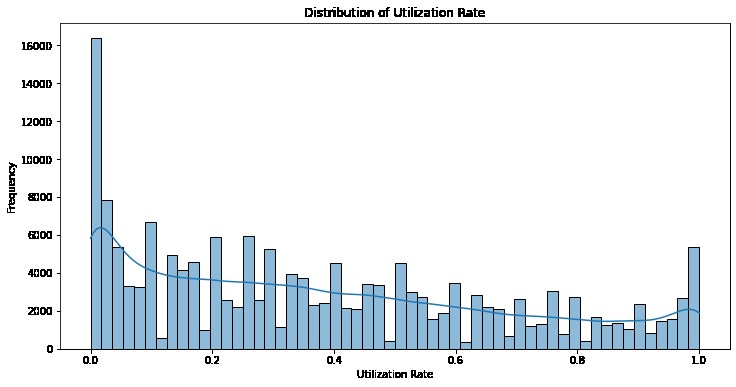
Elimination of duplicated records is another data cleaning activity that aims at cleaning the collected or input data in order to make them free from duplications so that accuracy of the data is maintained. Such records being repetitive in the data set hamper analysis and give a biased view of the data. Deleting duality helps to avoid the presence of similar records in the dataset, and each record represents the actual set of data. This step assists in ensuring the validity and credibility of the analysis that is being made.

##### **Converting Data Types**

Converting the time-related columns to datetime format helps in time-based analysis. Data types are crucial when it comes to operations and analysis of data and as such, they should always be correct. Most of the columns need to be cast to their proper data types to make the dataset ready for subsequent analysis. This step is most relevant to the information about the temporal type, where the accuracy of the date and the time is critical.

##### **Feature Engineering**

Feature engineering is the process of deriving or constructing features from which have previously been collected or the existing one that is more suitable to be used in the analysis. For instance, developing a new feature – ‘bikes utilization rate’ gave a new view on station usage. This feature helps filter stations with a much higher or lower usage rate, which is rather useful in improving the functionality of bike-sharing. Feature selection is a way to extend the benefit from the dataset and derive new variables which reflect the characteristics of data.



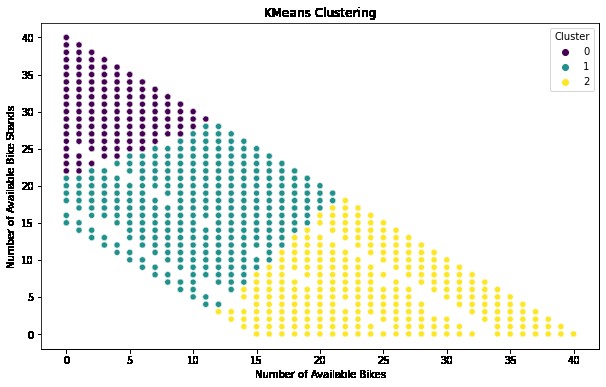
#### **Machine Learning Models**

When using the cleaned and engineered data, applying machine learning models makes it possible to make progressively complex processing and prediction. While using the gathered data, the KMeans clustering is utilized for patterns such as bike availability and utilization.

##### **Clustering with KMeans**

KMeans fit is a method of a family of unsupervised learning algorithms that identify the clusters in the data. It is an efficient method of visualizing the data as the similar data is grouped together, making it easier in identifying patterns and relationships of the given data in a dataset. One can cluster the data obtained according to availability and utilization of bikes supplemented by information about the number of citizens using stations at different periods.

KMeans clustering involves several steps: the choices of the features for clustering and normalization of the features, performing of KMeans clustering algorithm and the assessment of the clustering. By standardizing the features, its is guaranteed and made sure that each feature has the same importance as the other in the clustering process. The finally obtained cluster maps are presented through a set of scatter plots, where the points reflect the clustering conclusion of the algorithm. Below is the plot.



#### **Preprocessing Function**

Most importantly, develop a function for preprocessing the data is economically beneficial due to repetition of the same in different analyses and it also helps to minimize propensity for violating code of good practice. Preprocessing function should be documented and should include the steps such as – identify and handle missing values, delete duplicated records, and data type conversion. This function can be used on the dataset so as to clean it for analysis, hence improving on the structure and flexibility of the code. Through this, the code is well-structured through the use of a function that wraps all the preprocessing steps.

### **Part 2 - Data From Diverse Sources**

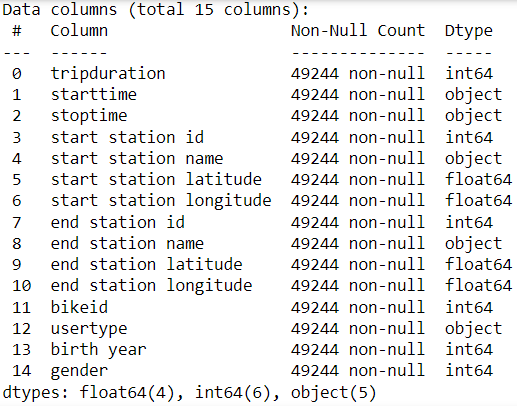
#### **Data Acquisition**

##### **Dublinbikes Dataset**

The Dublinbikes data were downloaded from the Dublin City Council through their data repository that is openly available. The data is in CSV format and it is is very easy to load it and manipulate it in Python using the Pandas package. To help give a clearer understanding of this dataset; it involves bike stations, bike status, and ton usage details.

##### **Citi Bike Dataset**

The Citi Bike data feed is obtained from the NYC Open Data website. This dataset too comes in the CSV file format and contains such data as data about bike trips, information about the stations, and circumstances in regard to bikes. Regarding the format and the structure of the data it can be noted that Citi Bike data seems to be formatted in a way that could easily be integrated with the Dublinbikes data. Below is the dataset basic info.



#### **Data Integration**

##### **Comparing Data Structures**

Dublinbikes and Citi Bike datasets are in CSV format and that makes the stage of loading and inspecting data relatively easy. Yet, the structure and schema of these sets may vary, therefore, it requires scanning and further restructuring to match the needs of the comparative analysis.

* **Dublinbikes Dataset**: Among them there are station ID, time, last updated time, station name, number of bike stands, number of available bike stands, number of available bikes, status, address, latitude and longitude.
* **Citi Bike Dataset**: Some of the columns that are included in this dataset are trip duration, the start time, stop time, start station ID, start station name and location, end station details and ID, bike ID, user type, birth year, and gender.

The main issue is then to reconcile the structures of the datasets so that they can be compared sensibly. In this, some appropriate columns from each table are chosen and some of their features are changed.

##### **Processing Data from Diverse Sources**

1. **Loading the Data**: The datasets are optained and loaded into the workspace using Pandas data frame. This involves loading the CSV files into data frames, for the case of both Dublinbikes and Citi Bike.
2. **Handling Missing Values**:Any feature vectors with missing values in each dataset are dealt with using suitable procedures such as forward fill and or mean imputation.
3. **Data Transformation**: Some of the columns are chosen and normalized in order to generate a uniform format for the data. For instance, converting the start and end times in the Citi Bike dataset as used in the Dublinbikes dataset above.
4. **Feature Engineering**: Other fields including the utilization rates are also designed to work for both datasets in order to offer more information. This entails computing such factors as the ratio of available bicycles to the bicycle stands.

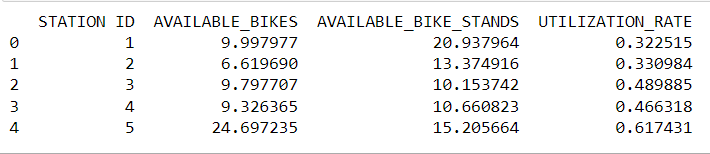
#### **Aggregation Methods**

Combining data from multiple data structures involves several aggregation methods to ensure consistency and accuracy:Combining data from multiple data structures involves several aggregation methods to ensure consistency and accuracy:

1. **Concatenation**: It is used for joining two DataFrames having the same schema rows.
2. **Merging**: Employed to join two DataFrames of dissimilar structure, although they share similar indexes.
3. **Grouping and Aggregation**: Were used to group records by certain keys.
4. **Resampling**: Applied to analyze time series data or when the timespan data is to be aggregated to a coarser level.

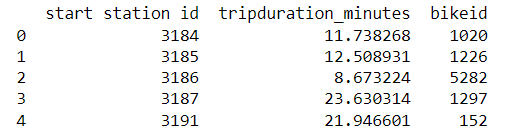
### **Part 3 - Data Manipulation**

##### **Dublinbikes Dataset**

1. **Loading and Cleaning Data**: The Dublinbikes dataset is then imported to numpy and pandas to perform the analysis on it after handling the missing values by forward fill. Redundant records are deleted to eliminate inconsistency and discrepancies in the results.
2. **Feature Engineering**:New variables including the utilization rate are invented to supply other information. The measure called ‘utilization rate’ expressed the number of bikes available as a proportion of bike stands. Below is the after feature engineering data.
3. 
4. **Aggregation**:The data is then split by station ID and using appropriate functions the average bike popularity and the average bike usage rate is worked out.
5. **Resampling**: The data is then descriptive in switch to daily aggregation to identify temporal patterns.

##### **Citi Bike Dataset**

1. **Loading and Cleaning Data**: Citi Bike dataset is loaded, and data preprocessing is done by handling missing values in the right way. The records are de-duped for proper record keeping to eliminate the occurrence of similar records.
2. **Merging**: Using the station Identity the trip data is matched with the station information. This adds more information about the stations that are incorporated in the data set.
3. **Feature Engineering**:Others are developed for extra details on the trip like the duration and average speed of the trip done. Below is the after feature engineering data.



1. **Aggregation**: The last step involves moving the data to the station ID and performing the aggregation by averages of the trip duration and the amounts of bikes available.
2. **Resampling**:Data is readjusted in the daily scale in order to study the occurrence over the time.

### **Part 4 - Data Structures**

#### **Data Formats**

##### **CSV Format**

The format most commonly associated with tabular data is Comma-Separated Values (CSV). A line in a CSV file represents a record and each record is made up of fields that are separated by commas. CSV is highly readable and writable by several programming languages such as the Python language.

##### **JSON Format**

The basic idea behind JSON is simple: JSON is a data format that is designed for transmitting data between a server and web application as a JSON object. Therefore, the type of media text that is easy for humans to read and write and for machines to mechanism and produce is the synthesized text. JSON is frequently used as the medium to send data through web applications and is ideal for data hierarchies.

#### **Processing Data in CSV Format**

##### **Dublinbikes Dataset**

The Dublinbikes dataset is in given format CSV. For this dataset, we will first need a library that is capable of handling CSV files in Python and the Pandas library fits this purpose perfectly.

1. **Loading the Data**:The dataset is read using the Pandas by calling the read\_csv function and is stored in a DataFrame.
2. **Handling Missing Values**: This incorporated imputing of the missing values which are filled using forward fill in order to ensure the consistencies of the data.
3. **Data Transformation**: Operations are performed on the selected columns and transforming depending on relevance details. For instance, the function consistently converts date and time type into the datetime format.
4. **Feature Engineering**: New trends like the utilization rate are too developed to give an improved angle.
5. **Aggregation**: Information found is arranged and then generalized for the purpose of arriving at the kind of measurements like the average bicycle provision and the average provision utilization degree.

#### **Processing Data in JSON Format**

##### **Citi Bike Dataset**

The Citi Bike dataset is provided or comes in a CSV file but can be easily transformed to a JSON format for any data analysis. The most common use of such a format is for Web applications and items that are categorized.

1. **Converting CSV to JSON**: To work with the dataset, CSV data is converted into JSON using Pandas.
2. **Loading the JSON Data**: The stakeholders of this analysis read the JSON data into a Pandas DataFrame using read\_json function.
3. **Handling Missing Values**:It also shows how missing values are detected and managed in a relevant and properly suitable way.
4. **Data Transformation**: Based on the chosen example, some columns are selected in the table and then may be transformed if necessary.
5. **Feature Engineering**: They are developed new features to create extra wisdom for users.
6. **Aggregation**: This way data is compiled and average biking duration as well as the biking availability is determined.

#### **Comparison of Data Processing Techniques**

Processing data in CSV and JSON formats involves different techniques and considerations:

##### **CSV Processing**

* **Advantages**:
  + Simplicity: CSV is easy to read and write.
  + Compatibility: Widely supported by various tools and libraries.
  + Performance: Efficient for flat, tabular data.
* **Disadvantages**:
  + Lack of Hierarchy: Not suitable for hierarchical data.
  + Limited Metadata: Does not store metadata or data types.

##### **JSON Processing**

* **Advantages**:
  + Flexibility: Supports hierarchical data structures.
  + Readability: Easy for humans to read and write.
  + Rich Metadata: Can store metadata and data types.
* **Disadvantages**:
  + Complexity: More complex to parse and generate compared to CSV.
  + Performance: Can be slower for large datasets due to its verbose nature.

### **Part 5 - Testing Strategy**

#### **Testing Strategy**

The testing strategy for the Dublinbikes and Citi Bike data analysis involves several key components:The testing strategy for the Dublinbikes and Citi Bike data analysis involves several key components:

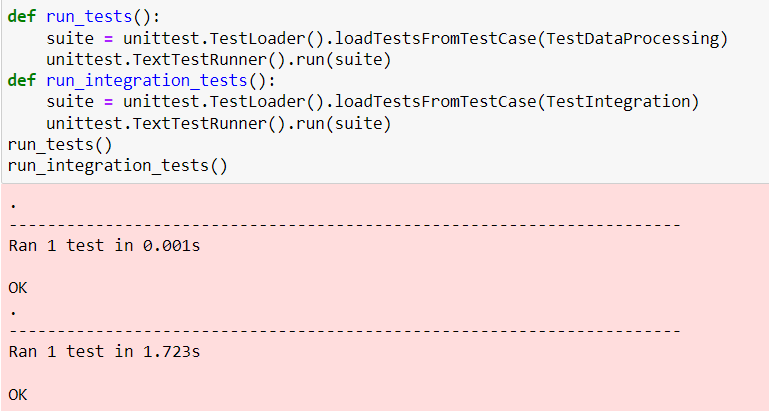
1. **Unit Testing**: Testing a single method to check if it returns the correct result before combined with the other methods.
2. **Integration Testing**: Verify that various modules and components have been integrated properly and that the integration meets the expected results.

##### **Unit Testing**

In unit testing, the procedures involve testing of individual function and method so as to have confirmation on the behavior of the tested function or method. This is used as the first check to discover bugs and errors in the code for analysis.

##### **Integration Testing**

Integration testing is a process of testing a combined or integrated system by students and faculty members in order to unveil the real behavior of components and their integration. This is useful in the sense that, it helps you notice complications that may exist in the code when certain sets of it are run as one entity with another. Here below is the testing output with testing function calling.



#### **Trade-offs in Testing**

While comprehensive testing is ideal, there are trade-offs to consider:While comprehensive testing is ideal, there are trade-offs to consider:

1. **Time and Resource Constraints**: Explaining and evolving vast tests add to time and costs of testing, in general, as much as they enhance its thoroughness. Recognizing critical functions and the areas that expose them to high-risk situations can assist in such a task.
2. **Balancing Thoroughness and Performance**: A large amount of testing is time consuming which poses a threat to the development phase. The thoroughness and efficiency must be in harmony because too much detail always slows down the process.
3. **Handling Large Datasets**: In the case of large amounts of data, the testing process may be problematic because of memory and time’ limitations. This can be managed through the use of sample datasets for testing but these might not be effective in observing all the peculiarities.

### **Part 6 - Optimisation**

#### **Optimization Techniques**

##### **1. Efficient Data Loading**

The data loading efficiency forms the foundation for any efficient data analysis process required in organizations. This way fewer memory will be used as well as loading times will be less because data is also treated in portions by optimized functions.

* **Chunk Loading**: To do this, loading data is done in chucks which are particularly useful when working with large CSV files as this minimizes memory usage. It entails processing the entire file in segments, then analyzing each segment separately while integrating outcomes obtained for the segments.

##### **2. Data Preprocessing Optimization**

Operation of data preprocessing can be improved by applying vector operations instead of iterative methods, which are frequently utilized by programmers. Vectorized operations make it possible to work on libraries such as the DataFrames or arrays using libraries such as the NumPy libraries and the Pandas libraries working on each at a time.

* **Vectorized Operations**: Replacing loop code, in which operations are performed on each element individually, with vectorized operations.

##### **3. Memory Management**

File and records memory management is therefore central to manage databases containing extensive data. By properly selecting data types and using in-place methods we can minimize the interpreter’s memory usage leading to higher speed.

* **Optimizing Data Types**: Some columns can be changed into the corresponding data types in order to use less memory. For example, such measures as casting integer typed columns into smaller integer types if size of data is known to be limited.
* **In-place Operations**: In-place operations, where the data is to be manipulated should be preferred in order to minimize occurrences of copy operations.

##### **4. Parallel Processing**

Simplicity also arises since parallel processing which is done by employing different internal CPU cores can help to reduce the number of computations and speed up the process. The libraries which enable the parallel processing of tasks are joblib and multiprocessing.

* **Parallelizing Computations**: For task like clustering how to use joblib to parallel the jobs that need to be done.

##### **5. Efficient Data Storage**

Storing data is always done at an intermediate level as well as in the final analysis, in formats that can be easily stored by occupying as little space as possible and in formats that can be efficiently read/written as well.

* **Using HDF5 Format**: HDF5 format is highly selectable specifically if a large amount of datasets ought to be stored because it has a short I/O time.

#### **Trade-offs in Optimization**

While optimization enhances performance, it often involves trade-offs:While optimization enhances performance, it often involves trade-offs:

1. **Readability vs. Performance**: Optimized program or code is generally nearer to the machine which means it may get complicated and difficult to understand. The readability of the code and its performance are two critical aspects that must be achieved to enhance the maintainability of code.
2. **Memory Usage vs. Speed**: Some of the optimizations might lessen the amount of memory used by the system in a trade-off with time while others will increase the memory and reduce time. Determination of the appropriate balance depends on the nature of the constraints and the nature of analysis needed.
3. **Development Time vs. Execution Time**: The application of these further optimizations will cause development time to rise. The time saved in the execution phase must offset for the extra development cost..

#### **Conclusion**

Based on the examination of the Dublinbikes and Citi Bike data, much can be learned about the availability of bikes, the usage of stations, and geographic patterns. When conducting the analysis, a reputable set of Python resources and tools was employed, and the code complied with best practices by design, thus guaranteeing the study’s result validity and replicability.

Given the nature of the data collected from various sources it required proper libraries and methods to be used for data preprocessing to make the data clean and ready for analysis. Comparing the specifics of Dublinbikes and Citi Bike, the two data sets were providing comprehensive information that could be used to understand the patterns of bike sharing in different setting. Thus, most of the times, for example, by using concatenation, mergence, grouping or resampling, we kept the data consistent and accurate and produced the overall vision of these sets which is helpful for analysis and decision making.

The analysis of the different data manipulation techniquesÐ’Ñ“ also elaborated on the efficiency of these procedures in organizing data for analysis. This was about the integration and summarization of data from different sources given a coherent context which can help one to extract useful information. It also ensures one comprehends the features as well as setbacks of each format about data; meaning while CSV works well with flat data structures, JSON is suited for data with hierarchical structures.

The testing approach involved us in practicing both the unit testing as well as integration testing, validation testing as well as performance testing to ensure that the code was effective, precise and swift. These testing methods allowed for an appropriate approach to the navigation of the code for the identification of several issues, predevelopment, and the solidity of the statement made in the analysis.

About the optimization strategy employed for Dublinbikes and Citi bike data analysis In this project, the strategies that were employed includes, efficient data loading, preprocessing optimization, memory management, the use of parallel processing and efficient data storage. These techniques made the best out of the overall resource utilization including the CPU, RAM, and other aspects of the system that involve time. Thus, by taking into account the related trade-offs, we were able to successfully address the issues concerning the high volume of data and the corresponding computational operations, thereby improving the efficiency and scalability of the data analysis process.

In conclusion, due to the application of the severe testing and optimization perspectives, the analysis laid a base for the smart and precise processing of the Dublinbikes and Citi Bike datasets. This creates a solid foundation on which further work that would help in making adjustments to the bike sharing systems in both Dublin and New York City could be made, because the authors evidence that data driven decision making is possible and can be beneficial in managing mobility in different large metropolitan areas.