**A Major-Project Report**
**on**

# Utilization-Aware Trip Advisor for Bike-sharing Systems

**Dissertation Submitted to the Dept. of Information Technology, SNIST**
**in the partial fulfillment of the academic requirements for the award of**

## B. Tech (Information Technology)

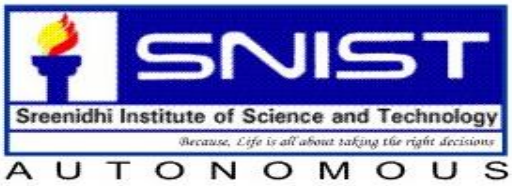
**under JNTUH**

**by**


## G. Jahanavi(16311A12C3)

## K.Ramya(16311A12D6)

## S.Sai Manohar(16311A12G2)



## Department of Information Technology
**School of Computer Science and Informatics**
**SreeNidhi Institute of Science and Technology (An Autonomous Institution)**
**Yamnampet, Ghatkesar Mandal, R. R. Dist., Hyderabad – 5o13o1**


**affiliated to**

## Jawaharlal Nehru Technological University Hyderabad
**Hyderabad – 500 085**

## 2019-2020

**A Major-Project Report**
**on**

# Utilization-Aware Trip Advisor for Bike-sharing Systems

**Dissertation Submitted to the Dept. of Information Technology, SNIST**
**in the partial fulfillment of the academic requirements for the award of**

## B. Tech (Information Technology)

**under JNTUH**
**by**

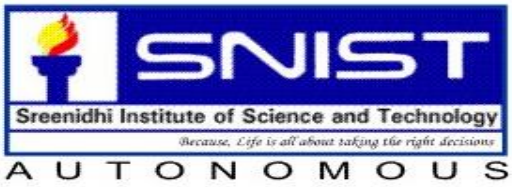

## G.Jahanavi(16311A12C3)

## K.Ramya(16311A12D6)

## S.Sai Manohar(16311A12G2)


**under the guidance**
**of**

## Ms. K.PriyaBhashini
**(Assistant Professor)**





## Department of Information Technology
**School of Computer Science and Informatics**
**SreeNidhi Institute of Science and Technology (An Autonomous Institution)**
**Yamnampet, Ghatkesar Mandal, R. R. Dist., Hyderabad – 5o13o1**


**affiliated to**

## Jawaharlal Nehru Technological University Hyderabad
**Hyderabad – 500 085**

## 2019-2020

# Department of Information Technology

**School of Computer Science and Informatics**
**SreeNidhi Institute of Science and Technology**
**(An Autonomous Institution)**
**(Affiliated to JNT University; ISO Certified 9001:2000)**
**Yamnampet, Ghatkesar, Hyderabad-501301**
**Ph.No. s: 08415-200597, 08415-325444, 9395533303**

# CERTIFICATE

This is to certify that the Dissertation entitled **"Utilization-Aware Trip Advisor for Bike-sharing Systems"** is a bonafide work done and submitted by **G.Jahanavi(16311A12C3), K.Ramya(16311A12D6), S.Sai Manohar (16311A12G2)** in the partial fulfilment for the award of B.Tech degree in Information Technology, **SreeNidhi Institute of Science and Technology,** Hyderabad, affiliated to **Jawaharlal Nehru Technological University Hyderabad (JNTUH), Hyderabad** is a record of bonafide work carried out by them under our guidance and supervision in the final year (second semester).

The results embodied in this Final year Major-Project work have not been submitted to any other University or Institute for the award of any degree or diploma.

Project Internal Guide

(Ms K Priya Bhashini)

Asst. Professor, Dept. of IT,
SNIST, Hyd.

Head of the Department

(Prof. V.V.S.S.S. Balaram)

Dept. of IT, SNIST, Hyd.

Project Coordinator

(Dr. P. Sreedhar)
Associate Professor, Dept. of IT,
SNIST, Hyd.

# DECLARATION

We, **G.Jahanavi (16311A12C3), K.Ramya (16311A12D6), S.Sai Manohar (16311A12G2)** students of **SreeNidhi Institute of Science and Technology, Yamnampet, Ghatkesar**, studying 4th year 2nd semester, **Information Technology** solemnly declare that the group project work, titled **"Utilization-Aware Trip Advisor for Bike-sharing Systems**" is submitted to **SreeNidhi Institute of Science And Technology** for partial fulfilment for the award of degree of Bachelor of Technology in **Information Technology.**

It is declared to the best of our knowledge that the work reported does not form part of any dissertation submitted to any other University or Institute for award of any degree.

# ACKNOWLEDGEMENTS

We would like to express our immense gratitude and sincere thanks to **Ms. K.PriyaBhashini**, Professor in Information Technology for **her** guidance, valuable suggestions and encouragement in completing the Major-Project work within the stipulated time.

We would like to express our sincere thanks to **Dr. P. Narasimha Reddy**, Executive Director**, Dr. T. Ch. Shiva Reddy**, Principal, Dr**. V. V. S. S. S. Balaram**, Professor & Head of the Department of Information Technology, **Dr. P. Sreedhar,** Associate Professor & Group-Project Work Coordinator of the Department of Information Technology, Sreenidhi Institute of Science and Technology (An Autonomous Institution), Hyderabad for permitting us to do our Group-Project work.

Finally, we would also like to thank the people who have directly or indirectly helped us and parents and friends for their cooperation in completing the Major-Project work.

**G. Jahanavi (16311A12C3)**
**K.Ramya (16311A12D6)**
**S.Sai Manohar (16311A12G2)**

# ABSTRACT

Over the past years, bike-distribution systems are growing in number and recognition in cities across the planet . Bike allocation system allows the users to rent bikes for trips. Thanks the increase in information technologies, it's easy for a user of the system to access a dock within the system to unlock or return bikes. Those technologies also yield a wealth of knowledge which will be want to explore how these bike-sharing organizations are utilised.

The hasty expansion ofbike-sharing arrangements has brought people humongous convenience. On the opposite hand, high transport resilience gives boost to problems for both users and operators. For users, vigorous distribution of shared bikes induced by uneven user demand often results in the check inor check out service unavailable at some stations. For operators, uneven bike usage comes with more bike broken and growing maintenance cost.

To solve these problems the existing prediction method given some recommendations to the system admin. They predicted that, in which perspectives most of bike rides happening. To do this they used machine learning algorithms and found the accuracy scores. Linear regression, decision tree and random forest to compare the accuracy scores.

In this project, with additional to that prediction method we will perform an exploratory data analysis on the datasets collected from Motivate, a company which provides bike-share system for many major cities in the United States. We will compare the system usage between three large cities: New York City, Chicago, and Washington, DC. We will also distinguish the rider ships among these three cities for those users that are registered or casual users.

# TABLE OF CONTENTS

# List of Figures

# List of Tables

# CHAPTER 1: INTRODUCTION

# *Chapter 1*

# *Introduction*

With the development of the economy, pollution and destruction caused by human activities to the natural environment were becoming more and more severe in recent years, and sustainable development has therefore become a consensus of the international community. In this circumstance, bike-sharing systems are developed as a replacement for short vehicle journeys due to its low pollution, low energy consumption and high flexibility. In addition to the reduction of need for personal vehicle trips, public bike-sharing systems can extend the reach of transit and walking trips, providing people with a healthy transportation option. With bike-sharing systems, a user can easily rent a bike with a smart card at a nearby station and return it at another station.

## 1.1 Problem Definition

For stations, the user demand is ever-changing and unbalanced, which often leads to the check in or check out service unavailable at some stations and has a negative impact on user experience. For bikes, the usage frequency of each bike is unevenly distributed, posing a problem for both riders and system operators. on the one hand, due to the high flexibility of bike sharing system, the system typically ends up with an uneven distribution of bikes across the different stations(due to the uncontrolled, uneven demand), often rendering the check in or check out service unavailable at some stations where bicycle docks are either fully occupied or empty. During peak periods, user demand characteristics differ among stations in certain areas. For example, rental demand usually gets larger in workday morning near residential areas, whereas return demand gets larger near commercial districts.

## 1.2 Motivation

The development of bike-sharing system has brought people enormous convenience. on the other hand, high transport flexibility gives rise to problems for both users and operators. For users, dynamic distribution of shared bikes caused by uneven user demand often leads to the check in or check out service unavailable at some stations. For operators, unbalanced bike usage comes with more bike broken and growing maintenance cost.

As a solution, the existing prediction method gave some recommendations to the system admin. They predicted that, in which perspectives most of bike rides happening, by finding the accuracy scores. By using, machine learning algorithms like linear regression, decision tree and random forest, to compare the accuracy scores.

By this admin can only know that which is the better algorithm that can be used for prediction, but not the further details like:

1) What is the average time duration that can be took for the rides?

2) Which city has the maximal number of rides? Which city has the maximalproportionof rides made by subscribers? Which city has the maximalproportionof rides made by short-term customers?

3) What is the average trip length for each city? Within that city, which type of user takes longer trips on median?

The above problems can be solved by using exploratory analysis on the collected data and by using some concepts of machine learning and some statistical computations.

### 1.3 Existing System

For stations, the user demand is ever-changing and unbalanced, which often leads to the check in or check out service unavailable at some stations and has a negative impact on user experience.

At present, operators perform bike redistribution based on monitor video and user complaints. However, this method has exposed the serious lag. It is usually when service unavailable events occur that operators start to give some scheduling instructions. When the vehicle arrives, service unavailable events may have passed for some time, which makes it difficult to meet the needs of users at rush hour.

For this, later they used simple prediction method to find the accuracy scores by using machine learning algorithms for prediction.

### 1.4 Limitation of Existing System

Due to the high supplenessof bike sharing system, the system typically ends up with an uneven distribution of bikes in different stations. During peak periods, user demand characteristics differ among stations in certain areas. For example, rental demand usually gets larger in workday morning near residential areas, whereas return demand gets larger near commercial districts.

The existing prediction method can only find the better algorithm by comparing their accuracy scores, but not fills the other details of riding.

### 1.5 Proposed System

In this project, in additional to that prediction method we will perform an exploratory analysis on data provided by Motivate, aorganization which provides bike distribution for large number of major cities in the United kingdom. We will compare the system usage between three large cities: New York City, Chicago, and Washington, DC. We will also compare the patterns of each system for those customers who are, regular users and casual users. And also find the duration time of the trip and proportions of trips and other details.

# CHAPTER 2: LITERATURE SURVEY

# *Chapter 2*

# *Literature Survey*

## 2.1 Introduction

In this section of the related work we describe the existing system, the limitation of the existing system, the earlier version, current version, proposed version, and expected results of this project.

## 2.2 Relevant Paper Analysis

**2.2.1 Title:** Mobility Modeling and Prediction in Bike-Sharing Systems

**Year**: 2016

**Author:** Ji Hu, Yuanchao Shu, Zidong Yang, Peng Cheng, Jiming Chen, Thomas Moscibroda

**Description**

As an innovative mobility scheme, public bike-distribution has grown adequately worldwide. Though providing convenient, low-cost and environmental-friendly transportation, the unique features of bike-distribution organizations grant raise to problems to both users and operators. The primary issue among these problems is the uneven distribution of bicycles caused by the ever-changing usage and (available) supply. This bicycle imbalance issue necessitates efficient bike re-balancing strategies, which depends highly on bicycle mobility modelling and prediction. In this paper, for the first time, we propose a spatio-temporal bicycle mobility model based on historical bike-sharing data, and devise a traffic prediction mechanism on a per-station basis with sub-hour granularity. We extensively evaluated the performance of our design through a one-year dataset from the world's largest public bike-sharing system (BSS) with more than 2800 stations and over 103 million check in/out records. Evaluation results show an 85 percentile relative error of 0.6 for both check in and checkout prediction. We believe this new mobility modelling and prediction approach can advance the bike re-balancing algorithm design and pave the way for the rapid deployment and adoption of bike-sharing systems across the globe.

Shared transportation has grown tremendously in recent years as a result of the rise of the sharing economy and growing environmental, energy and economic concerns. Among the various forms of shared-use mobility1, public bike-sharing systems (BSS) have become increasingly popular with a significant growing presence over t he past decade. Available figures indicate that there are more than 500 bike-sharing programs running in at least 49countries with one million shared bikes in 2015.

**2.2.2Title:** Data-Driven Utilization for Bike-Sharing Systems

**Year:** 2017

**Author:** Ji Hu, Zidong Yang, Yuanchao Shu, Peng Cheng, Jiming Chen

**Description**

Despite high convenience and flexibility, a notable problem in bike-sharing systems is unbalanced bike usage, which means a small part of bikes are used much more frequently than others. Bikes that are used too much are vulnerable and hence increase repair bills and lead to potential service denied. In 2012, the very first bicycle from Hangzhou bike-sharing system became a permanent exhibit in the Low-Carbon Technologies Museum in China. This bicycle is reported to be rented for over 6,000 times and ridden for more than 20,000 kilometres in 3 years. Similarly, the most tireless bicycle from 2016 has been rented for 5,616 times, over 15 times on average each day.

Intuitively, operators can balance bike usage by leading users to use those unpopular bikes based on usage counts of each bike. However, leading users to rent a specific bike is not practical. Based on our analysis on real bike-sharing dataset from Hangzhou, we observe that bikes located in some stations are much more likely to be used and moved to another active station. Hence, by introducing the station property of activeness, we transform the original problem of picking bikes to recommending check-in and check-out stations. By using the proposed trip advisor, we aim to guide users to ride bicycles between stations with different levels of activeness, therefore avoiding circumscribed circulation among active stations. For users, an advisor can not only help them choose stations with adequate bicycles, but also ensure a higher success rate when returning bikes. Also, different incentive mechanisms can be leveraged to better prompt the balancing process.

In this paper, we propose a trip advisor that recommends the optimal pair of stations to rent and return bikes. Through guiding the actions of users, it can help balance bike usage, reduce operation cost and enhance user experience. Firstly, to make sure users can find bikes and available lockers, success rates of rental and return should be predicted for each station. Different from traditional demand prediction methods, we present probabilistic forecast methods on a minute timescale instead of predicting the exact stock number on sub-hour granularity. Secondly, in order to balance bike usage through station recommendation, a station property must be associated with bike usage frequency. We define activeness for each station by exploiting the idea of Page Rank. These two parts constitute the core content of the trip advisor framework.

**2.2.3Title**: Exploring trip characteristics of bike-sharing system uses

**Year**: 2018

**Author**: Lei Kang,Yi-Hsuan Wu, Yu-Ting Hsu, Po-Chieh Wang

**Description:**

The rapid growth and expansion of bike-allocation organization in late years have drawn rising analysis interest's toaimmense amount of case studies which acknowledge a diversity of development courses. Imposition-side scrutiny is one of the focuses in the relevantarticle, as the awarenessof user aspecton a public transportation system is demanding for quality decision-making with regard to system planning and operation. The land-use pattern and associated activity system can be the fundamental factors of travel demand generation in the context of urban transportation planning; it has been recognized that well-connected streets, mixed land uses, and close proximity to retail activities can increase the propensity of cycling propose a GIS-based approach which integrates these factors into an optimization problem to determine the locations of bike-sharing stations.

The records of electronic transactions of bike-sharing systems, including the locations and timestamps of bike rentals and returns, facilitate data-driven research on system performance and demand analysis at an aggregate level.

The existing and future usage of CitiBike system (the bike-sharing system serving New York City and Jersey City) based on the spatiotemporal patterns of interactions by modeling the arrival and departure rates of bikes at rental stations.

The population and commercial activities in small cities may not have enormous impact on the usage of public bikes. However, they suggest that focusing on dense and walk able areas, providing quality bicycle facilities, and partnering with schools and companies should let the system solidly competitive in a small area. Irrespective the scale of a city, the essence is to suit the local's cycling conditions.

other than the demand-inducing factors, facilities such as bike lanes that improve safety and comfort of cycling environments are identified to be influential to user's willingness to ride bikes.

The share of riding bikes to work is still considerably small in most cities. In order to encourage more people to cycle in their commuting trips to confront the climate change and public health, Cole-Hunter et al. (2015) suggest that the presence of greenness in the surroundings of work/study area and the availability of public bike stations near home have positive effects on willingness to commute by cycling.

# CHAPTER 3: SYSTEM ANALYSIS

# *Chapter 3*

# *System Analysis*

### 3.1 INTRODUCTION

The purpose of software requirements specification specifies the intentions and intended audience of the SRS. The scope of the SRS identifies the software product to be produced, the capabilities, application, relevant objects etc. Software Requirements Specification: It's a description of a particular software product, program or set of programs that performs a set of function in target environment. The SRS contains the details of process, functions of the product, user characteristics. The non functional requirements if any are also specified. The remaining section of the SRS specifies the functionality of these systems. Further the need of Interfaces is also described in the next part of SRS. The criteria for the non-functional requirements, the constraints on the system and assumptions and dependencies (if any) are also described in the remaining sections. 3.2 Software Requirement Specification.

### 3.2 Software Requirement Specification.

A software requirements specification (SRS) is a description of a software system to be developed.

Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers (in market-driven project, these roles may be played by the marketing and development divisions) on what the software product is to do as well as what it is not expected to do. Software requirements specification permits rigorous assessment of requirements before design can begin and reduces later redesign.

### 3.2.1 SYSTEM SPECIFICATIONS

- **Hardware Requirements**:

RAM: 4GB and Higher

Processor: Intel i3 and above

Hard Disk: 5ooGB: Minimum

- **Software Requirements:**

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification.
**Operating System** : Windows or Linux
**Python IDLE** : python 2.7.x and above
jupyter notebook.
Setup tools and pip to be installed for 3.6 and above
**Language** : Python Script

- **Python Packages**

Python packages required to run this application:

**a) Numpy**

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the basic package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object.
- Sophisticated (broadcasting) functions.
- Tools for integrating C/C++ and Fortran code.
- Useful linear algebra, Fourier transform, and random number capabilities.

**b) Matplotlib**

Matplotlib is a Python 2D plotting library which producesquality figures in a varietyof hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

**c) Datetime**

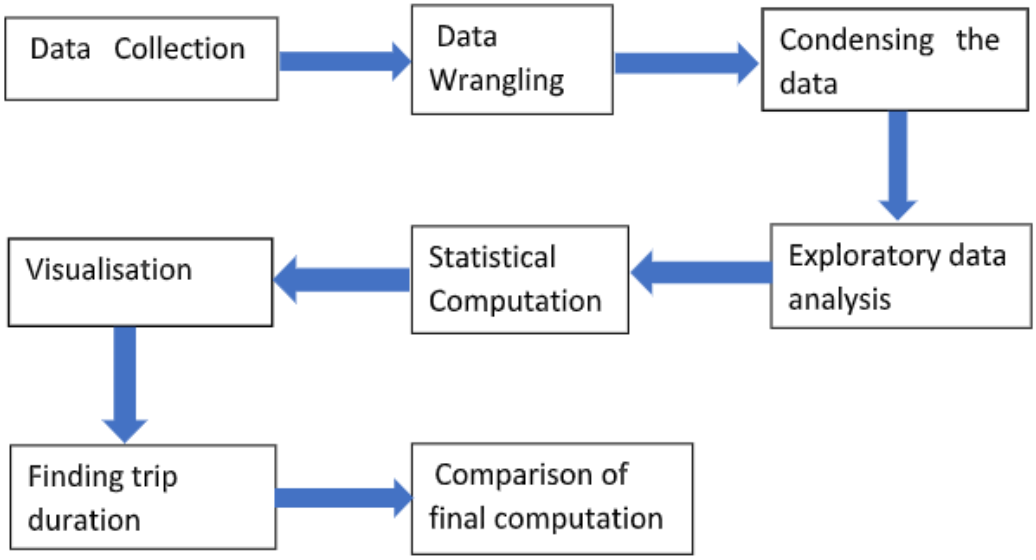
Date and time are not a data type of its own, but a module named datetime can be imported to work with the date as well as time. Datetime module comes built into Python, so there is no need to install it externally.

Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.

d) **Pprint**

The pprint module provides a capability to "pretty-print" arbitrary Python data structures in a form which can be used as input to the interpreter. If the formatted structures include objects which are not fundamental Python types, the representation may not be loadable.

### *3.3 Architecture:*



## 3.4 Modules Description

### 3.4.1 Data wrangling

Data Wrangling is the process of converting and mapping data from its raw form to another format with the purpose of making it more valuable and appropriate for advance tasks such as Data Analytics and Machine Learning.

This may include further munging, data visualization, data aggregation, training a statistical model, as well as many other potential uses. Data munging as a process typically follows a set of general steps which begin with extracting the data in a raw form from the data source, "munging" the raw data using algorithms (e.g. sorting) or parsing the data into predefined data structures, and finally depositing the resulting content into a data sink for storage and future use.

### 3.4.2 The goals of data wrangling:

Reveal a “deeper intelligence” within your data, by gathering data from multiple sources
Provide accurate, actionable data in the hands of business analysts in a timely matter
Reduce the time spent collecting and organizing unruly data before it can be utilized
Enable data scientists and analysts to focus on the analysis of data, rather than the wrangling
In this project, we will focus on the record of individual trips taken from selected cities: New York City, Chicago, and Washington, DC. If we visit the datasets. Some data wrangling of inconsistencies in timestamp format within each city has been performed.

### 3.4.3 Condensing the trip data

It should also be observable from the datasets that three cities provides contrasting information. Even where the information is the same, the column names and formats are sometimes different. To make things as simple as possible when we get to the actual exploration, we should trim and clean the data. Cleaning is used to consistent the data formats of cities, while trimming focuses only on the parts of the data we are most interested in to make the exploration easier to work with.

We will generate new data files with five values of interest for each trip: trip duration, starting month, starting hour, day of the week, and user type. Each of these may require additional wrangling depending on the city:

Duration: This has been given to us in seconds (New York, Chicago) or milliseconds (Washington). in our project we are going to convert these duration into the format of minutes.

Month, Hour, Day of Week: Ridership volume is likely to change based on the season, time of day, and it maybe a weekday or weekend. Use the start time of the trip toobtain these values. The NYC data includes the seconds in their timestamps, while Washington and Chicago do not.

User Type: The users have different patterns in choosing the riderships. These patterns are based on the user utilization characteristic. In Washington the users are divided into two formats: ’Registered’ and ‘Casual’. In New York and Chicago users are of ‘Subscriber’ and

'Customer'. For consistency, we need to convert the Washington labels to match the other two.

The condensed data file for each city should consist only the data fields indicated above.

### 3.4.4 Exploratory data analysis

Exploratory Data Analysis (EDA) is an approach to analysing data sets to summarize their main characteristics, by using visualisation methods. A statistical model can be used or not, but primarily EDA is for seeing what the data can tell us beyond the formal model. Exploratory data analysis was promoted by John Tukey to encourage statisticians to explore the data, and possibly formulate hypotheses that could lead to new data collection and experiments. EDA is different from initial data analysis (IDA), which focuses more narrowly on checking assumptions required for model fitting and handling missing values and making transformations of variables as needed. EDA encompasses IDA.

# CHAPTER 4: DESIGN

# *Chapter 4*

# *Design*

### 4.1 Introduction

Design Engineering deals with the various UML [Unified Modeling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process in which we translate the requirements into representation of the software. Design is the place where quality is rendered in software engineering. Design is the means to accurately translate customer requirements into finished product.

### 4.2 Unified Modeling Language Description:

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

**User Model View**

This view represents the system from the user perspective.

The analysis representation describes a usage scenario from the end-users perspective.

- **Structural model view**

In this model the data and functionality are arrived from inside the system.

This model view models the static structures.

- **Behavioral Model View**

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

- **Implementation Model View**

In this model the structural and behavioral parts of the system are built.

### 4.3 Use case Diagram:

This diagram is considered as a behavioural diagram. The purpose of use case is to present overview of the functionality provided by the system in terms of actors, their goals.

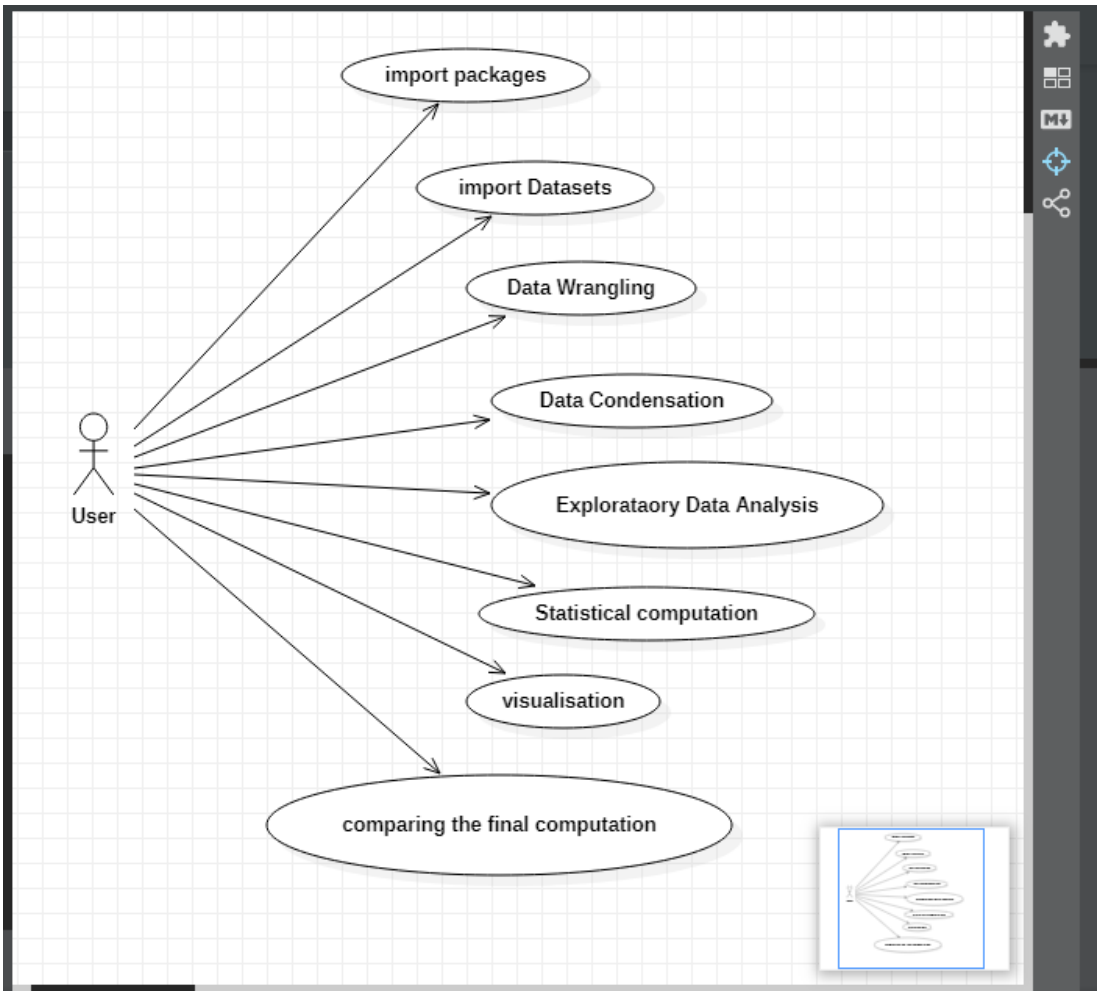



Fig: Use case diagram

**4.4 Class Diagram:**

This diagram considered as structural diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. Private visibility hides information from anything outside the class partition.

Public visibility allows all other classes to view the marked information

A class is having three sections. The upper part holds the name of the class and the middle part contains the attributes of the class. The bottom part gives the methods or operations the class can take or undertake.

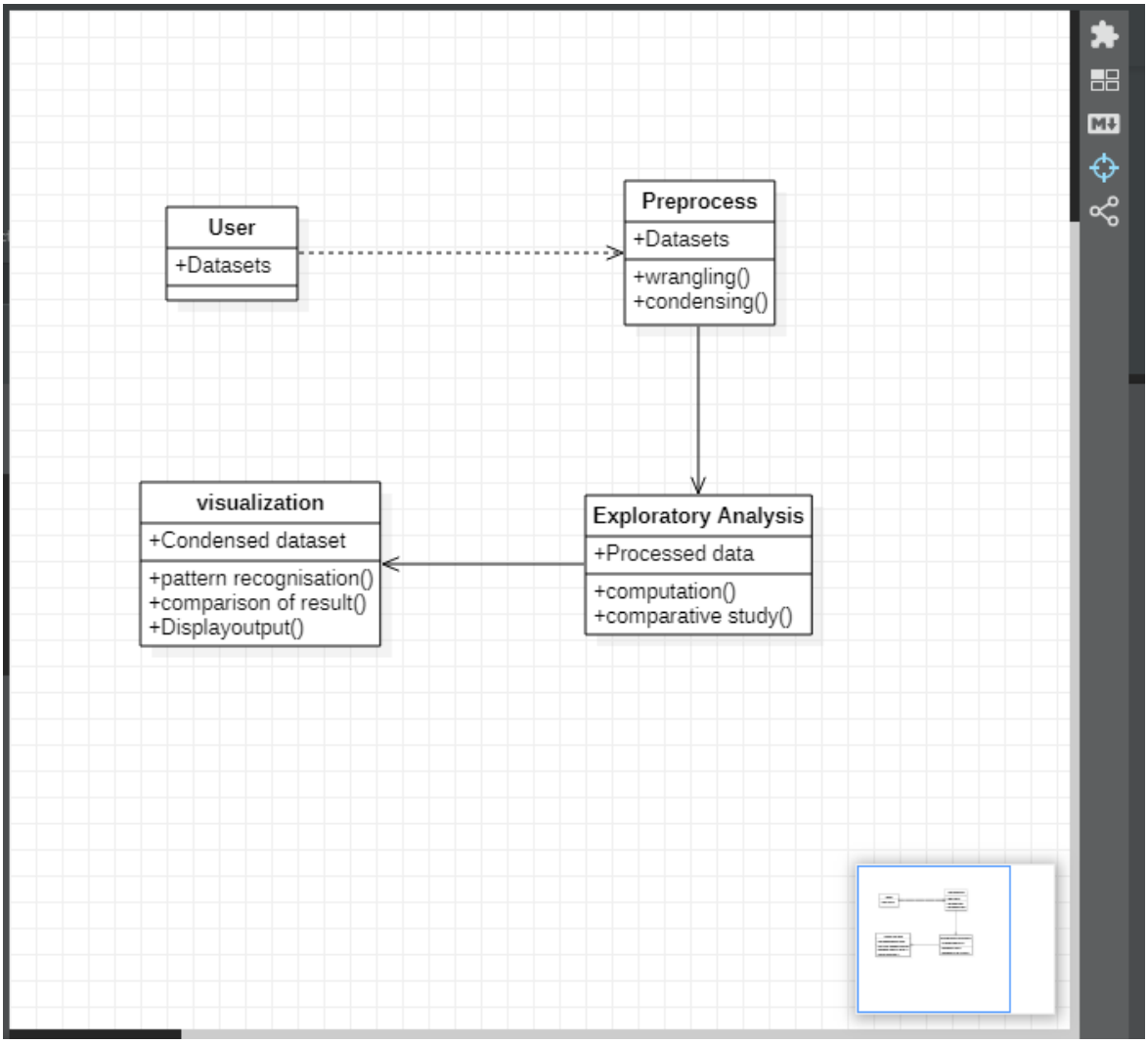



Fig: Class Diagram

### 4.5 Sequence Diagram:

This diagram is used to represent the flow f actions take place between the components of a system. It includes time representation to show the sequence of interaction between the components.

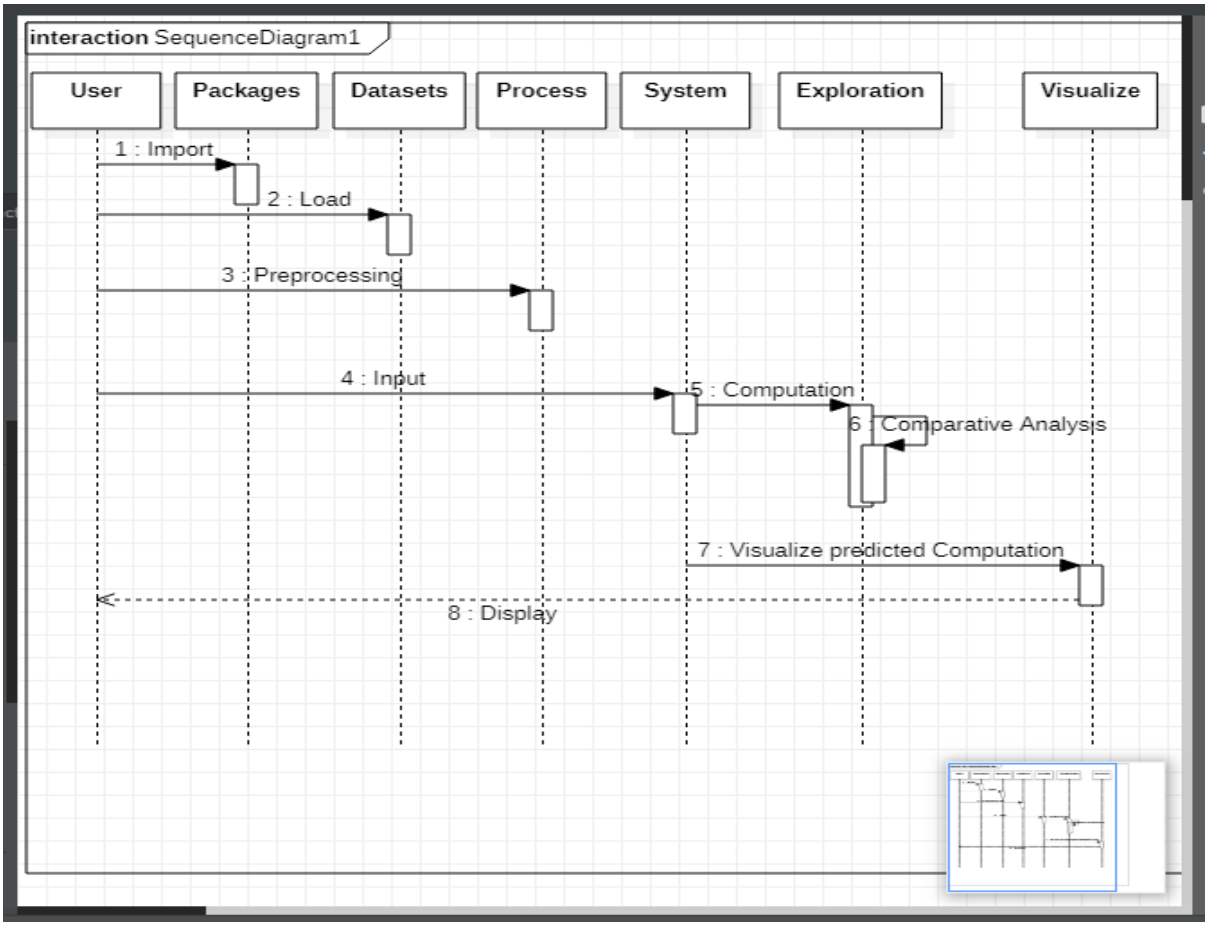



Fig: Sequence Diagram

**4.6 Collaboration Diagram:**

A collaboration diagram show the objects and relationships involved in an interaction, and the sequence of messages exchanged among the objects during the interaction. The collaboration diagram can be a decomposition of a class, class diagram, or part of a class diagram. It can be the decomposition of a use case, use case diagram, or part of a use case.

The collaboration diagram shows messages being sent between classes and object.

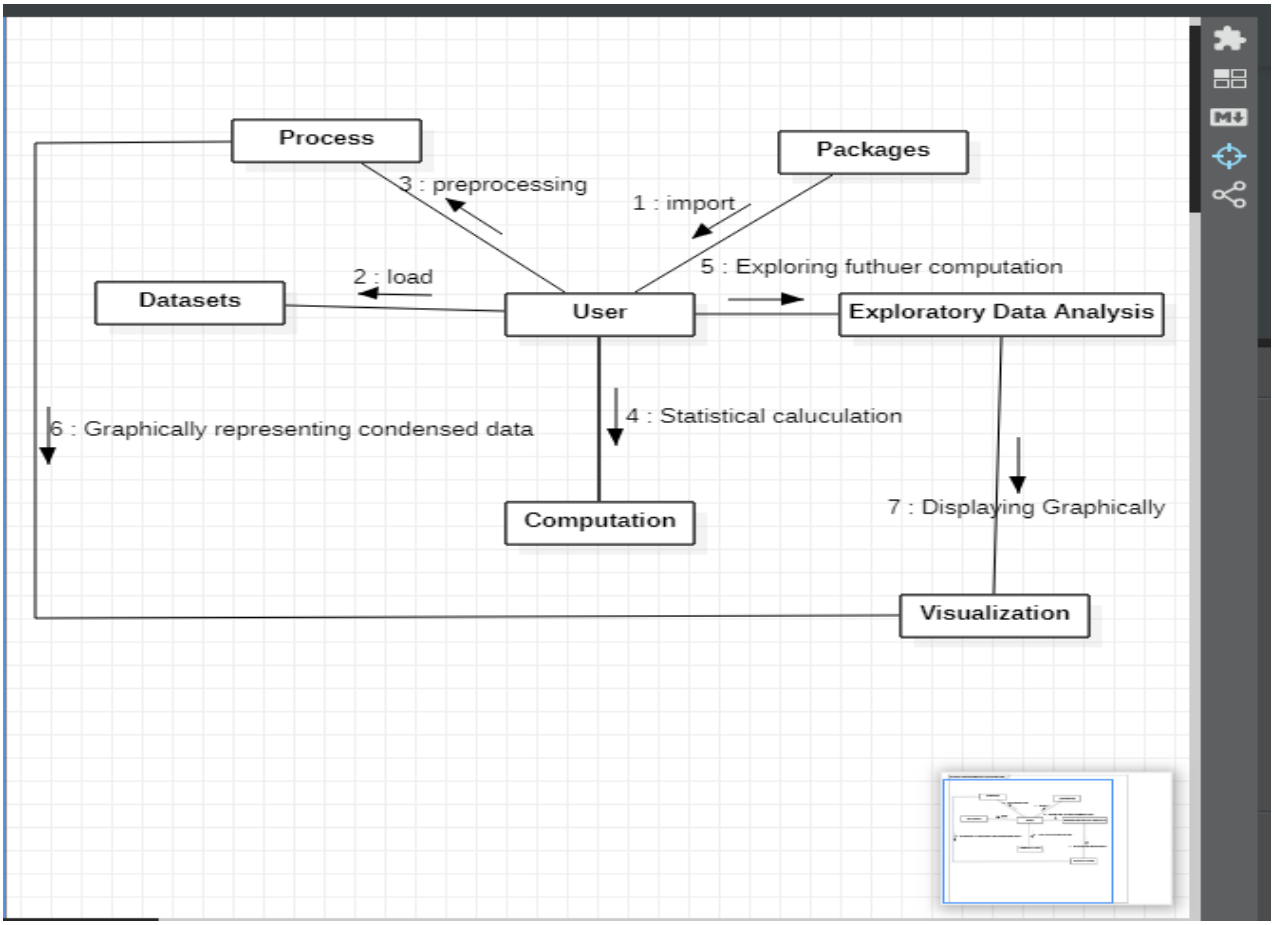



Fig: Collaboration Diagram

### 4.7 Activity Diagram:

Activity diagram are a loosely defined diagram to show the activities in a stepwise manner. UML, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system.
UML activity diagrams could potentially model the internal logic of a complex operation.

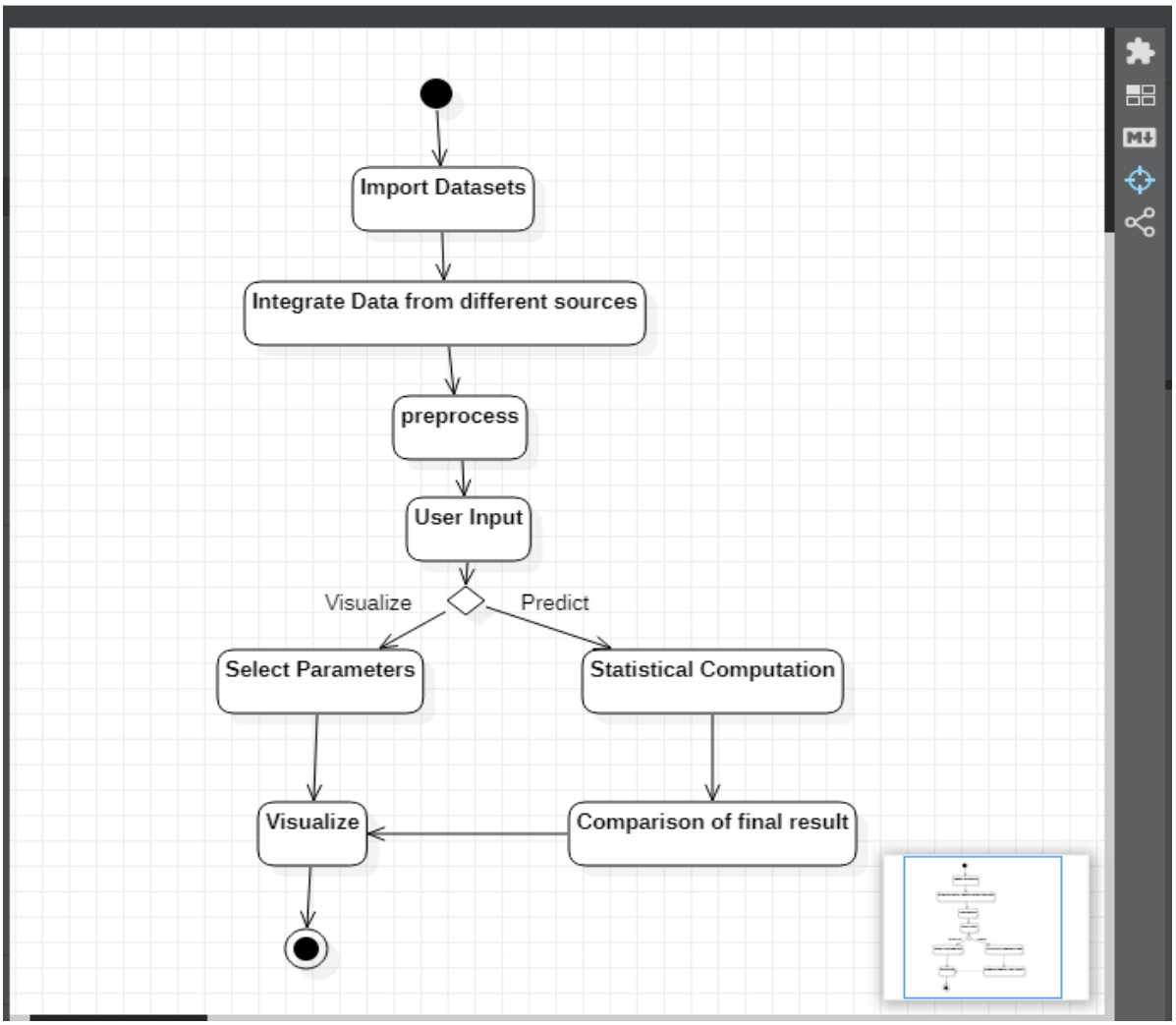



Fig: Activity Diagram

**4.8 State Chart Diagram:**

This diagram is an illustration of the states an object can attain as well as the transitions between those states in the Unified Modelling Language .In this context, a state defines a stage in the evolution or behaviour of an object, which is a specific entity in a program or the unit of code representing that entity.

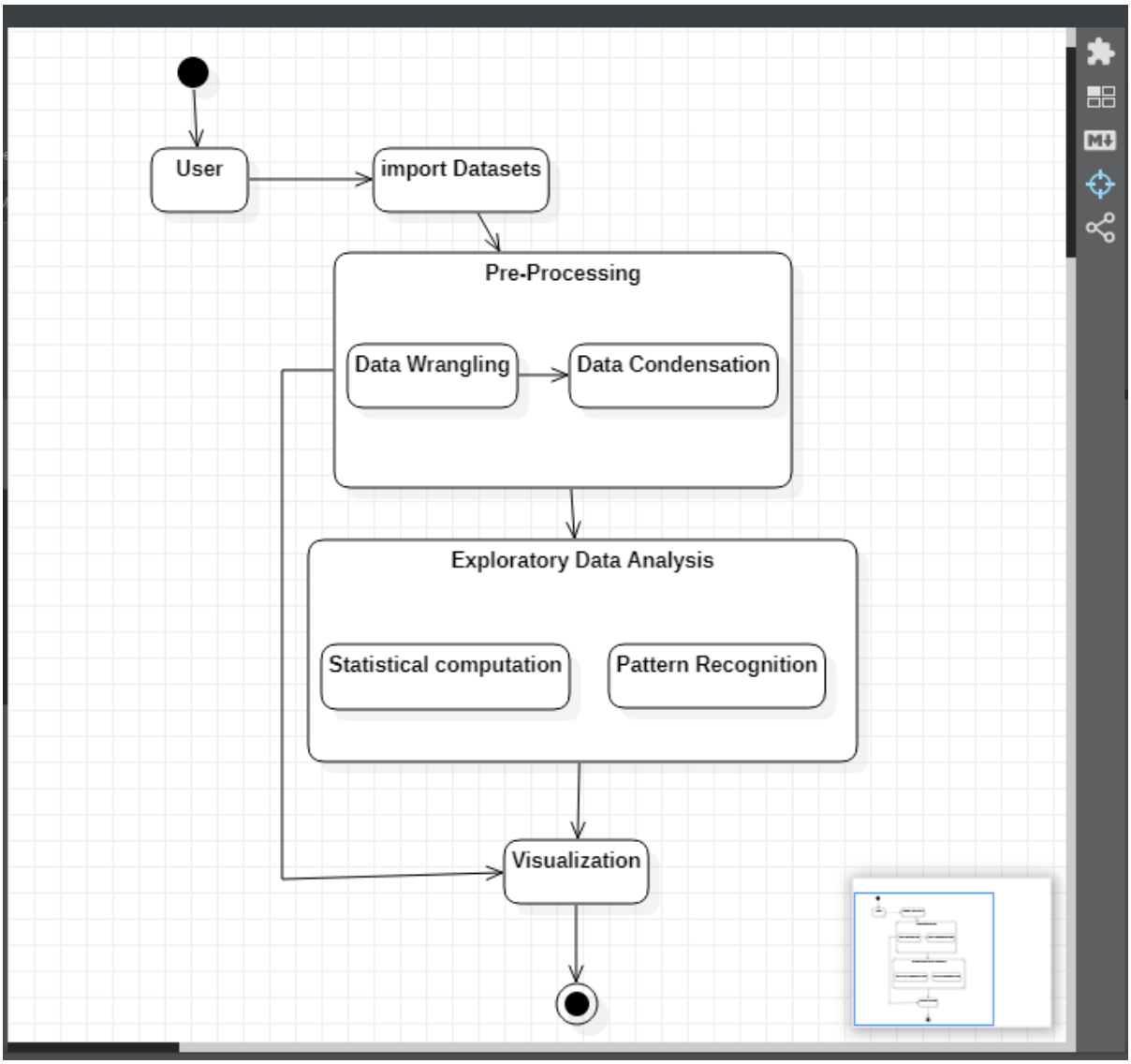



Fig: State Chart Diagram

**4.9 Component diagram:**

The component diagram main purpose is to show the structural relationships between the components of a system. A component represented implementation items, such as files and executables.

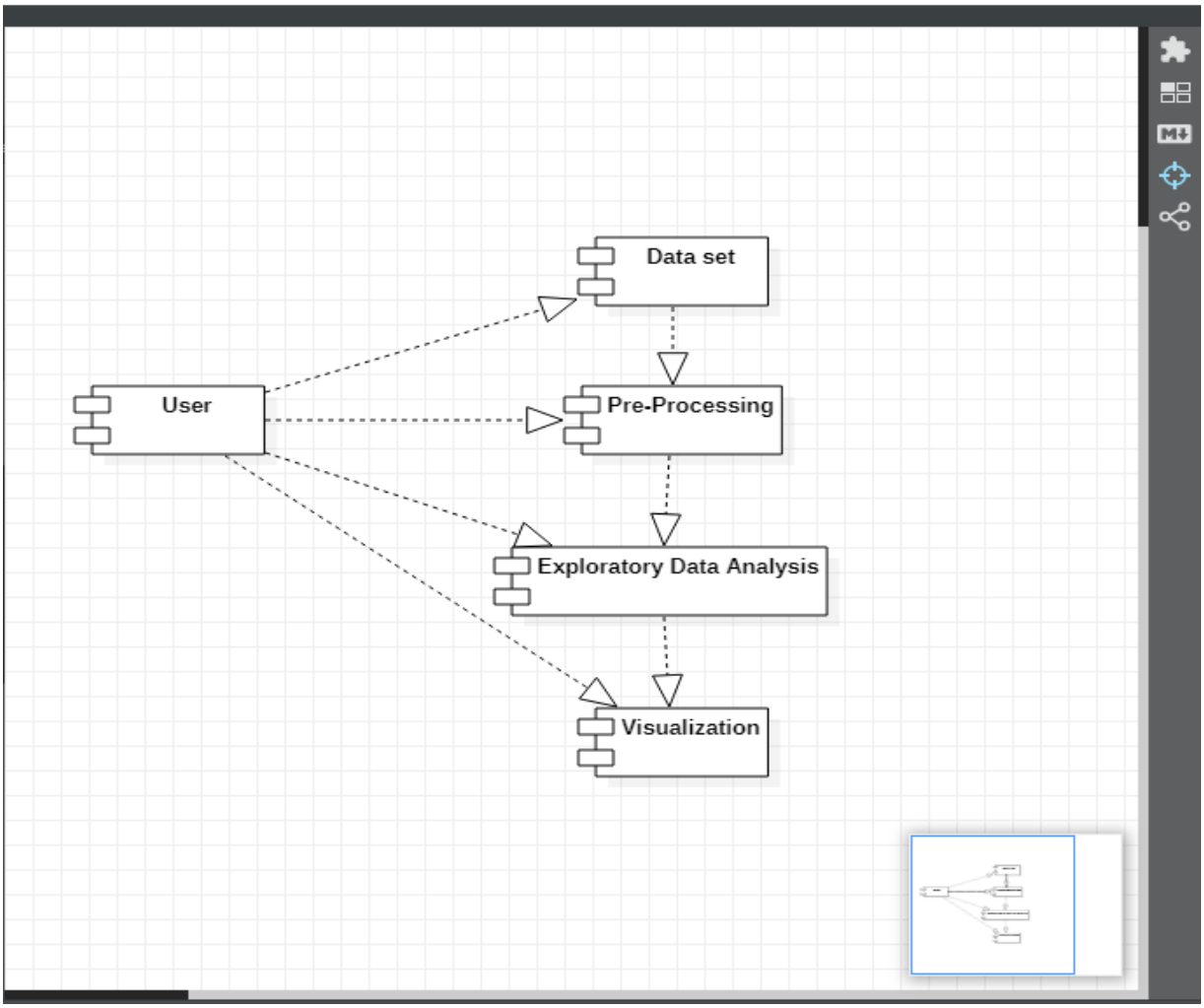



Fig: Component Diagram

# CHAPTER 5

# IMPLEMENTATION&RESULTS

# *Chapter 5*

# *Implementation & Result*

### 5.1 Statistical computation:

Now we have collected the data and applied wrangling and condensation on it; we must start exploring the data. In this we will write some code to compute statistical calculation from the data. For visualization purpose we use matplotlib module to create histograms of the data.

We should compute the details like, Which city has the maximal number of rides? Which city has the maximalproportionof rides made by subscribers? Which city has the maximalproportionof rides made by short-term customers?

In [8]:
```
def number_of_trips(filename):
    """
    This function reads in a file with trip data and reports the number of
    trips made by subscribers, customers, and total overall.
    """
    with open(filename, 'r') as f_in:
        # set up csv reader object
        reader = csv.DictReader(f_in)

        # initialize count variables
        n_subscribers = 0
        n_customers = 0

        # tally up ride types
        for row in reader:
            if row['user_type'] == 'Subscriber':
                n_subscribers += 1
            else:
                n_customers += 1

        # compute total number of rides
        n_total = n_subscribers + n_customers

        # return tallies as a tuple
        return(n_subscribers, n_customers, n_total)
```

```
data_file = {'Chicago': './data/Chicago-2016-Summary.csv', 'NYC': './data/NYC-2016-Summary.csv', 'Washington': './data/Washington-2016-Summary.csv'}
for city in data_file:
    n_subscribers, n_customers, n_total = number_of_trips(data_file[city])

    p_subscribers = round(n_subscribers / n_total *100, 2)
    p_customers = round(n_customers / n_total *100, 2)

    summary = "{}: {} subscribers ({}%), {} customers ({}%), {} in total".format(city, n_subscribers, p_subscribers, n_customers, p_customers, n_total)
    print(summary)
```

```
Chicago: 54982 subscribers (76.23%), 17149 customers (23.77%), 72131 in total
NYC: 245896 subscribers (88.84%), 30902 customers (11.16%), 276798 in total
Washington: 51753 subscribers (78.03%), 14573 customers (21.97%), 66326 in total
```

Now, we have to continue our further work, by choosing one city. Within that city, we should know which type of user takes longer rides on average: Subscribers or Customers?

To do this we should complete some more statistical work, we will use some functions in the code to compare the rides of Subscribers and Customers.

Then the output will be:

```
filename = './data/Chicago-2016-Summary.csv'
avg_subscribers, avg_customers = compare_subscriber_customer(filename)
summary = "In Chicago, the average riding length of subscribers is {} minutes, whereas that of customers is {}
g_subscribers,2), round(avg_customers,2))
print(summary)
```

```
In Chicago, the average riding length of subscribers is 12.07 minutes, whereas that of customers is 30.98.
```

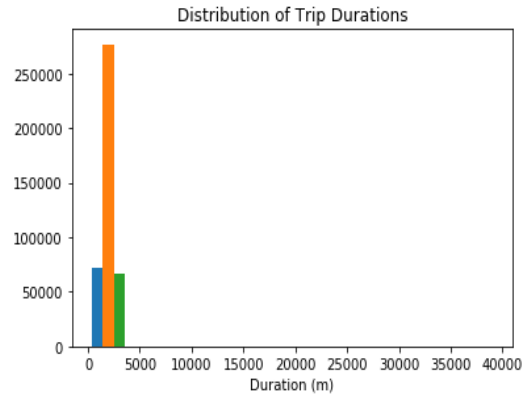
### 5.2 Visualisation

Now we are going to use some functions like.hist() this function performs the computations and creates plotting objects for creating a histogram, but the plot is actually not accomplished until .show() function is executed. The .title() and .xlabel() functions provide some labelling for plot context.

We will now use these functions to create a histogram of the trip times for the city we selected. We are not separating the Subscribers and Customers for now: just collecting all of the trip times and plotting them.

```
data_files = ['./data/Chicago-2016-Summary.csv', './data/NYC-2016-Summary.csv', './data/Washington-2016-Summary.csv']
data = []
for data_file in data_files:
    data.append(retrive_data(data_file))

plt.hist(data)
plt.title('Distribution of Trip Durations')
plt.xlabel('Duration (m)')
plt.show()
```

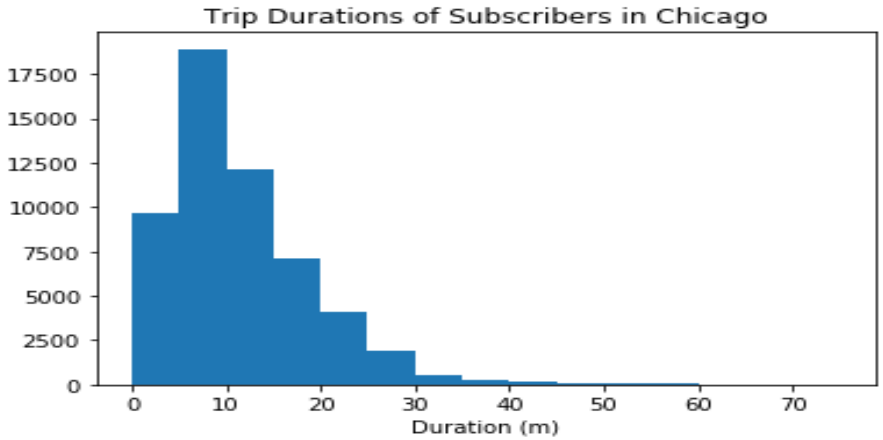


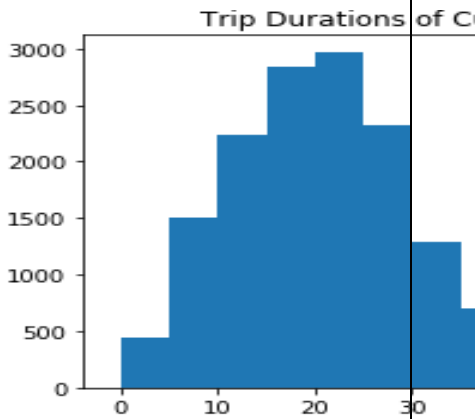
Now we are using some parameters of the .hist() function to plot the distribution of trip times for the Subscribers in our selected city. We have to do the same thing for only the Customers and also adding limits to the plots so that only trips of duration less than 75 minutes are plotted.

For each group, we have to find where is the peak of each distribution?

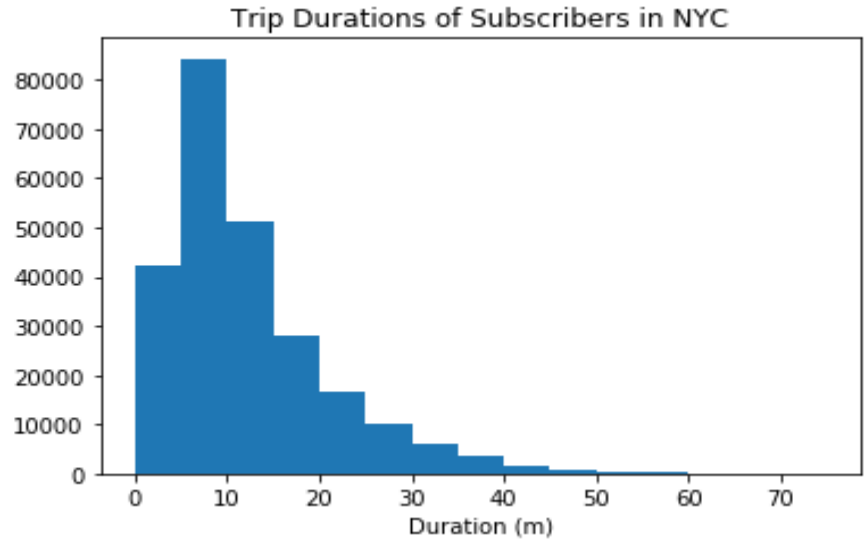
output plots will be:

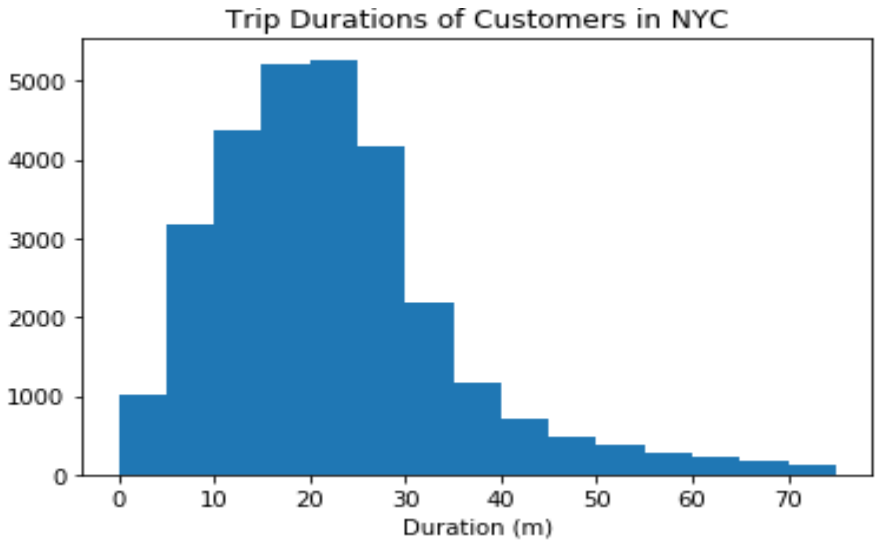
In Chicago, the peak of subscribers' trip time is 5-10 minutes, whereas that of customers' is 20-25.



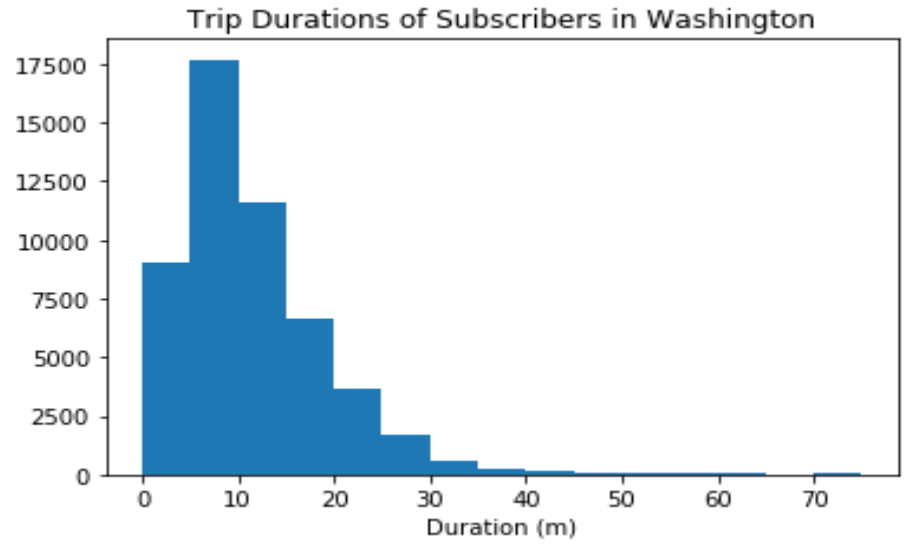



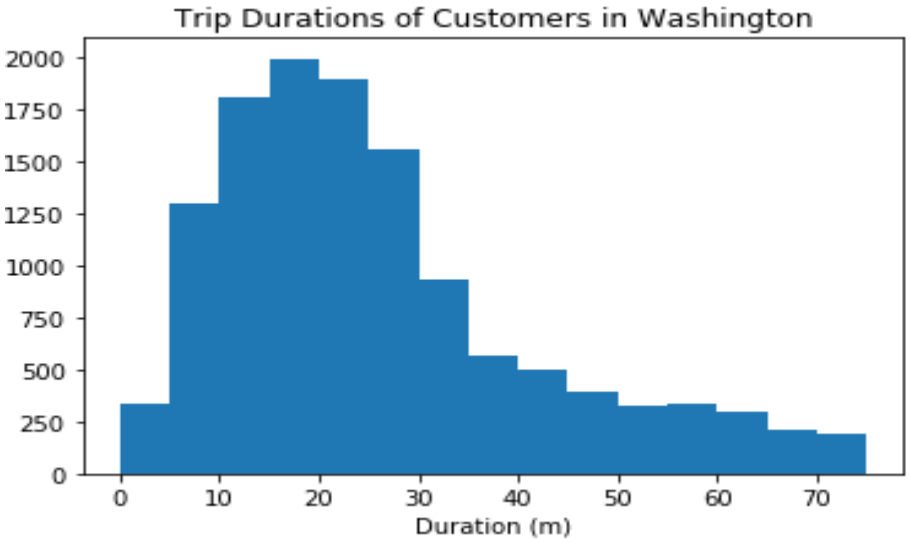

In NYC, the peak of subscribers' trip time is 5-10 minutes, whereas that of customers' is 20-25.




In Washington, the peak of subscribers' trip time is 5-10 minutes, whereas that of customers' is 15-20.




We have performed an initial exploration into the collected data. we have compared the relative volume of trips made between three U.S. cities and the ratioof trips made by Subscribers and Customers. For one of these cities, we have investigated differences between Subscribers and Customers in terms of how long a typical trip lasts.

As final computation for this project we are going to find some more information like,

We are finding the patterns of two types of customers on weekly basis and some more information regarding them.

By all this work we are going to add some more detailed information to the existing prediction method. This can be very useful for understanding various differences of ridership in each and every city or place that we will select.

## 5.3 SAMPLE CODE

In this section of the documentation we describe the important part of the coding which is being used in the proposed project. The complete code is not included in the report, only the sections which are most important are included in this section of the report.

***Code for data wrangling***

```
import csv

from datetime import datetime

f rom pprintimport pprint

def print_first_point(filename):

trip_reader=csv.DictReader(f_in)

first_trip=trip_reader.__next__()

pprint(first_trip)

return(city,first_trip)

example_trips={}

fordata_fileindata
```

***code for condensing the data***

```
def duration_in_mins(datum,city):

if' tripduration' in datum:

duration=float(datum['tripduration'])

else:

duration=float(datum['Duration (ms)'])/1ooo

duration=duration/6o

return duration

if 'Customer 'in datum:

user_type=datum['Customer']

elif 'usertype' indatum:

user_type=datum['usertype']

else:

user_type=datum['Member Type']
```

```
if user_type=='Registered':
user_type='Subscriber'
elifuser_type=='Casual':
user_type='Customer'
return user_type
tests={'NYC':'Customer',
'Chicago':'Subscriber',
'Washington':'Subscriber'}
def condense_data(in_file,out_file,city):
    with open(out_file,'w')asf_out,open(in_file,'r')asf_in:
out_colnames=['duration','month','hour','day_of_week','usertyp
new_point={}
new_point['duration']=duration_in_mins(row,city)
new_point['month'],new_point['hour'],new_point['day_of_week
new_point['user_type']=type_of_user(row,city)
trip_writer.writerow(new_point)
```

**Code for statistic computation**

```
defriding_length(filename):
withopen(filename,'r')asf:
reader=csv.DictReader(f)
total_length=0
n_over30=0
n_total=0
forrowinreader:
duration=float(row['duration'])
total_length+=duration
n_total+=1
ifduration>30:
n_over30+=1
```

```
average_length=total_length/n_total

p_over30=n_over30/n_total

return(average_length,p_over30)

average_length,p_over30=riding_length(data_file[city])

average_length=round(average_length,2)

p_over30=round(p_over30*100,2)

summary="The average trip length for {} is {} minutes, with {}% people having a ride for more than 30 minutes.".format(city,average_len

print(summary)

defcompare_subscriber_customer(filename):

withopen(filename,'r')asf:

reader=csv.DictReader(f)


n_subscribers=0

length_subscribers=0

n_customers=0

length_customers=0


forrowinreader:

duration=float(row['duration'])

ifrow['user_type']=='Subscriber':

n_subscribers+=1

length_subscribers+=duration

else:

n_customers+=1

length_customers+=duration


avg_subscribers=length_subscribers/n_subscribers

avg_customers=length_customers/n_customers


return(avg_subscribers,avg_customers)
```

**Code for data visualising**

```
defretrive_duration(filename,user_type='Subscriber'):
data=[]
withopen(filename)asf:
reader=csv.DictReader(f)
for row in reader:
if row['user_type']==user_type:
duration=float(row['duration'])
if duration<75:
data.append(duration)
return data

def draw_duration_plot(data,city,user_type):
plt.hist(data,bins=15,range=(0,75))
plt.title('Trip Durations of {}s in {}'.format(user_type,city))
plt.xlabel('Duration (m)')
plt.show()
user_types=['Subscriber','Customer']

for city in data_files:
for user_type in user_types:
draw_duration_plot(retrive_duration(data_files[city],user_type),city,user_type)
```

***Code for pattern***

```
def separate_day_of_week(day_of_week,n_user,duration,length):
if day_of_week=='Monday':
n_user[0]+=1
duration[0]+=length
```

```
elif day_of_week=='Tuesday':
n_user[1]+=1
duration[1]+=length
elif day_of_week=='Wednesday':
n_user[2]+=1
duration[2]+=length
elif day_of_week=='Thursday':
n_user[3]+=1
duration[3]+=length
elif day_of_week=='Friday':
n_user[4]+=1
duration[4]+=length
elif day_of_week=='Saturday':
n_user[5]+=1
duration[5]+=length
elif day_of_week=='Sunday':
n_user[6]+=1
duration[6]+=length
return_user
def calculate_numbers_duration(filename,user_type):
with open(filename)asf:
reader=csv.DictReader(f)
n_user=[0,0,0,0,0,0,0]
duration=[0,0,0,0,0,0,0]
for row in reader:
if row['user_type']==user_type:
length=float(row['duration'])
separate_day_of_week(row['day_of_week'],n_user,duration,length)
returnn_user,duration
def calculate_avg_duration(n_user,duration):
```

```
avg_duration=[0,0,0,0,0,0,0]
for n in range(0,7):
avg_duration[n]=duration[n]/n_user[n]
return avg_duration

def draw_plot(data_subscribers,data_customers,y_label,city):
y1=data_subscribers
y2=data_customers
x1=range(1,8)
x2=range(1,8)

plt.plot(x1,y1,label='Subscribers',linewidth=3,color='r',marker='o',markerfacecolor='blue',mar
kersize=12)

plt.plot(x2,y2,label='Customers')

plt.title('{} in {} during the week'.format(y_label,city))
plt.xlabel('Day of Week')
plt.ylabel('{}'.format(y_label))

plt.legend()
plt.show()
n_subscribers=calculate_numbers_duration(f,'Subscriber')[0]
n_customers=calculate_numbers_duration(f,'Customer')[0]
draw_plot(n_subscribers,n_customers,'Number of people using bikes',city)

total_duration_subscribers=calculate_numbers_duration(f,'Subscriber')[1]
total_duration_customers=calculate_numbers_duration(f,'Customer')[1]

avg_duration_subscribers=calculate_avg_duration(n_subscribers,total_duration_subscribers)
avg_duration_customers=calculate_avg_duration(n_customers,total_duration_customers)
```

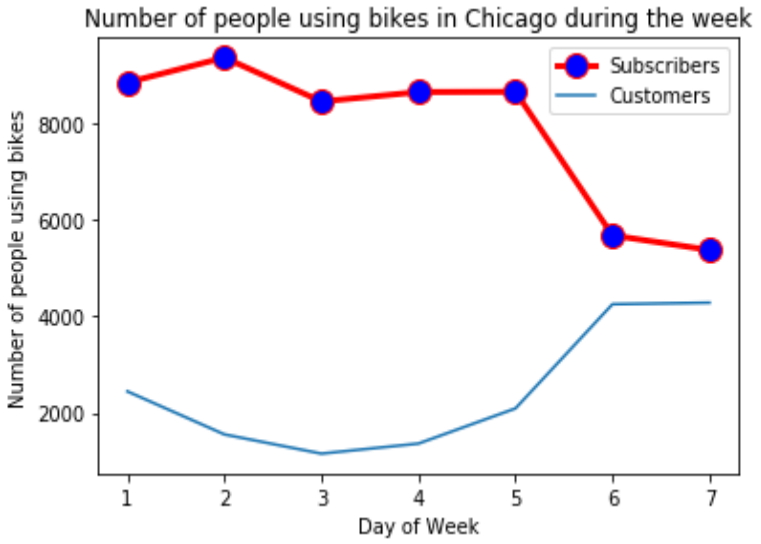
draw_plot(avg_duration_subscribers,avg_duration_customers,'Average duration of using bikes',city)

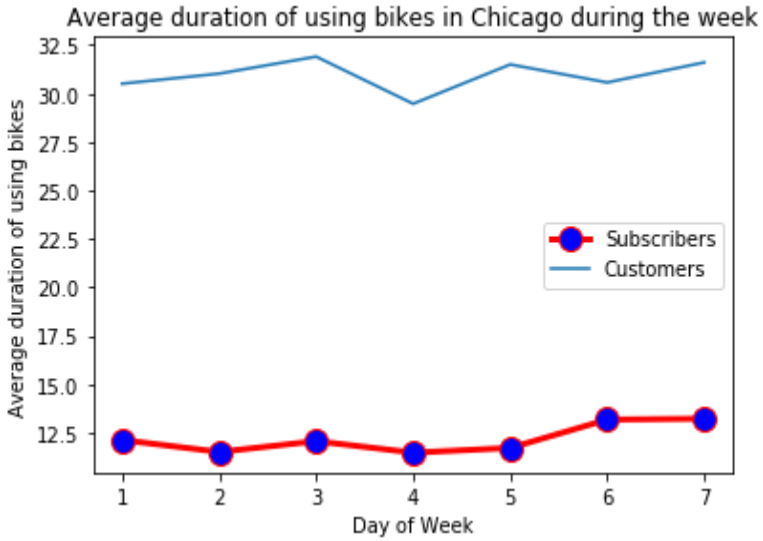
### 5.4 output screenshots

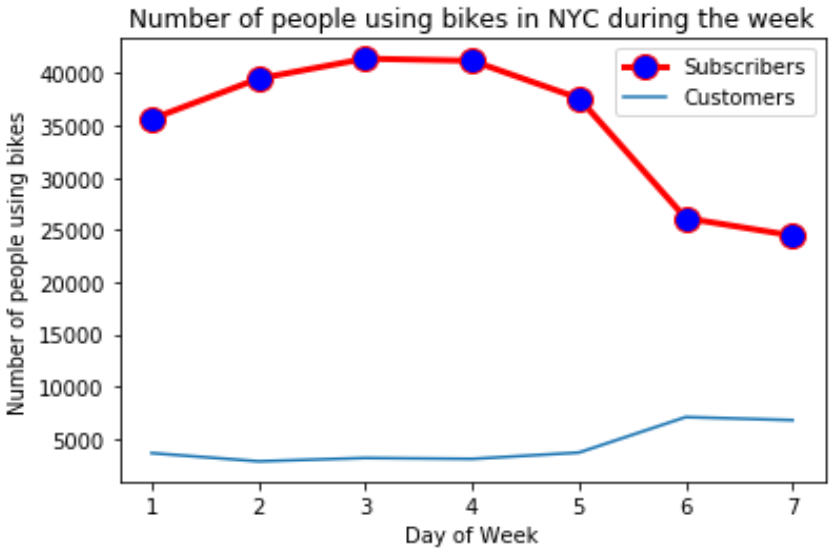
There are differences of the pattern of ridership between the weekends and weekdays.

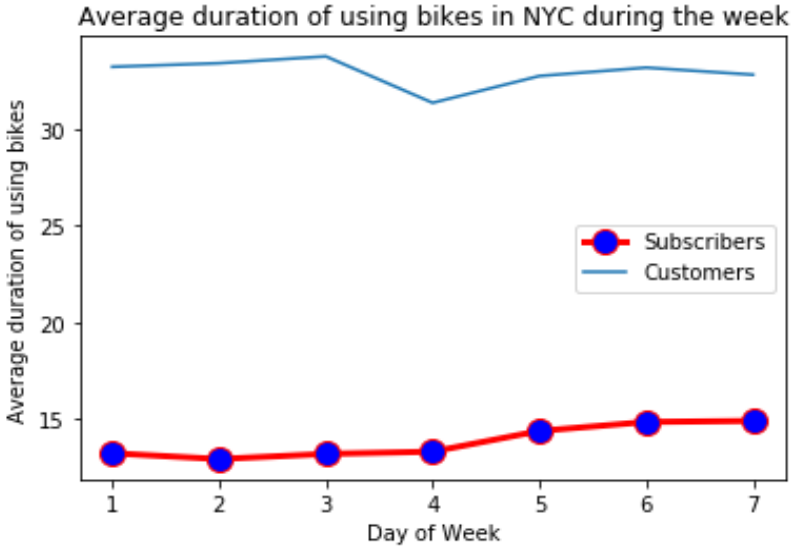
In all of the three cities, the subscribers prefer to use the system on weekdays, whereas the customers on weekends.

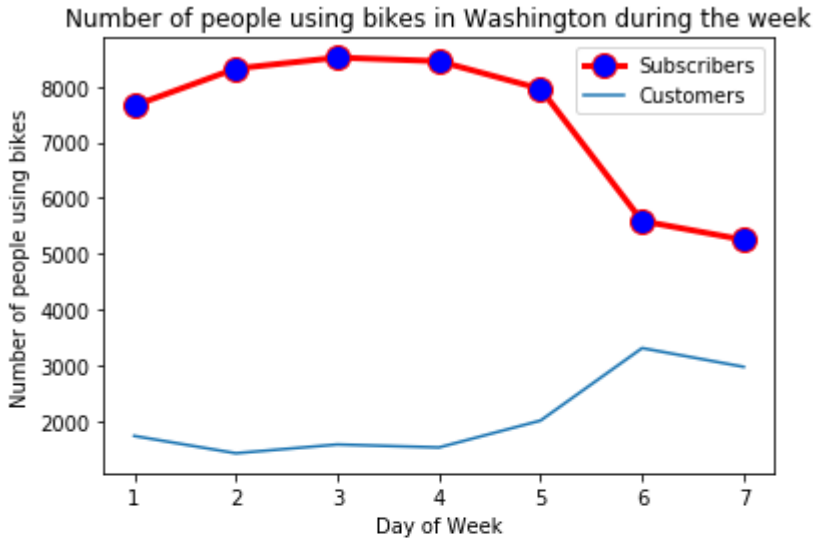
While the average riding duration of subscribers is around 15 minutes through the week, that of customers is about 30 minutes.

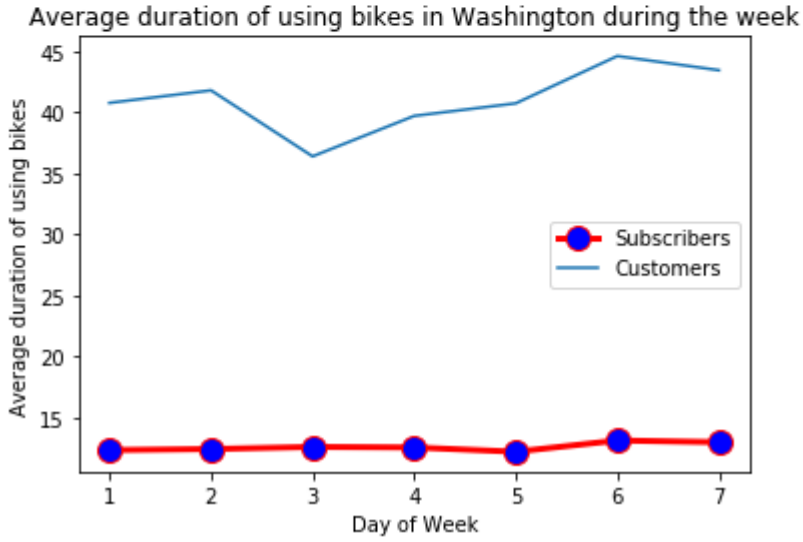



Number of people using bikes in NYC during the week
Subscribers
Customers
Number of people using bikes
40000
35000
30000
25000
20000
15000
10000
5000
1
2
3
4
5
6
7
Day of Week


Average duration of using bikes in NYC during the week
Subscribers
Customers
Average duration of using bikes
30
25
20
15
1
2
3
4
5
6
7
Day of Week


Number of people using bikes in Washington during the week
Subscribers
Customers
Number of people using bikes
8000
7000
6000
5000
4000
3000
2000
1
2
3
4
5
6
7
Day of Week


Average duration of using bikes in Washington during the week
Subscribers
Customers
Average duration of using bikes
45
40
35
30
25
20
15
1
2
3
4
5
6
7
Day of Week

# CHAPTER 6: TESTING &VALIDATION

# *Chapter 6*
# *Testing & Validati0n*

## 6.1 Introduction

Software testing is the process of verifying a system with the purpose of identifying any errors, gaps or missing requirement versus the actual requirement. Software testing is broadly categorised into two types - functional testing and non-functional testing.

**6.2 Types of Testing:**

**6.2.1 White Box Testing:** focused on the structure of the software

**6.2.2 Black Box Testing:** dose not focuse on the the inner structure of software

**6.2.3 Functional test:** Testing the functionalities of the software

**6.2.4 Unit Testing:**testing the small individual component of the system

**6.2.5 Integration Testing:** testing the integrated components**.**

## 6.3 Design of test cases and scenarios

6.3.1 Test Case for Data Wrangling

| Testcase Id | Test case Name | Testcase i/p | Expected o/p | Actual o/p | Pass/Fail | Remarks |
|---|---|---|---|---|---|---|
| 1 | Wrangling | Individual Data files | Details of first trip in files | First trip details | Pass | Testcase passed |
| 2 | Wrangling | Datafile with fault details | Details not rec0gnized | Error | Fail | Testcase Failed |

6.3.2 Test Case for Data Condensing

| Testcase Id | Test case Name | Testcase i/p | Expected o/p | Actualo/p | Pass/Fail | Remarks |
|---|---|---|---|---|---|---|
| 3 | Condensation | Wrangled datafile | Data with same format and column names | Condensed data | Pass | Testcase passed |
| 4 | Condensation | Wrong dataset | Unformatted data | Error | Fail | Testcase Failed |

6.3.3 Test Case for Statistical Calculation

| Testcase Id | Test case Name | Testcase i/p | Expected o/p | Actual o/p | Pass/Fail | Remarks |
|---|---|---|---|---|---|---|
| 5 | Statistical calculation | Condensed dataset | Correct results of c0mputati0n | Correct results of computation | Pass | Testcase passed |
| 6 | Statistical calculation | Wrong computational formula | Inc0rrect meth0d | Err0r | Fail | Testcase Failed |

6.3.4 Test Case for Data Visualization

| Testcase Id | Test case Name | Testcase i/p | Expected o/p | Actual o/p | Pass/Fail | Remarks |
|---|---|---|---|---|---|---|
| 7 | Visualization | Data with classified riders | Result in the form of histograms | Histograms | Pass | Testcase passed |
| 8 | Visualization | Data with incorrect type of riders | Customers type n0t tallied | Error | Fail | Testcase Failed |

6.3.5 Test Case for Data Pattern recognization

| Testcase Id | Test case Name | Testcase i/p | Expected o/p | Actual o/p | Pass/Fail | Remarks |
|---|---|---|---|---|---|---|
| 9 | Pattern of ridership | Duration, Usertype, N0_user | Plot diagram of recognized pattern | Pattern recognized | Pass | Testcase passed |
| 10 | Pattern of ridership | Function with wrong parameters | Wrong parameters | Error | Fail | Testcase Failed |

# CHAPTER 7: CONCLUSION

# *Conclusion*

In this project, in additional to that prediction method we will perform an exploratory analysis on data provided by Motivate, a BSS provider for plenty major cities in the US. We compared the system utilization between three large cities: New York , Chicago, Washington. We also see the differences within each system for those users that are registered and casual users. And also find the duration time of the trip and proportions of trips and other details.

And based on the analysis of general characteristics, spatial temporal patterns utilization in BSS, we propose a novel architecture of a utilization aware trip advisor to help the organizations which maintains bss in balancing the bike based on its utilization.

# *References*


[1] P. DeMaio, “Bike-sharing: History, Impacts, Models of Provision, and Future,” Journal of

Public Transportation DeMaio, pp.41–56, 2009. [online]. Available: http://www.transitinformatics.org/test/nctr/wp- content/ uploads/2o1o/o3/JPT12-4DeMaio.pdf

[2] P. Midgley, “Bicycle-sharing schemes: enhancing sustainable mobility in urban areas,” United Nations, Department of Economic and Social Affairs.

[3] J. Froehlich, J, and N. oliver, “Sensing and Predicting the Pulse of the City through Shared Bicycling,” in IJCAI, 2009.

[4] A. Kaltenbrunner, R. Meza, J. Grivolla, J. Codina, and R. Banchs, “Urban Cycles and Mobility Patterns: Exploring and Predicting Trends in a Bicycle based Public Transport System,” Pervasive and Mobile Computing.

[5] P. Vogel and D. C. Mattfeld, “Strategic and operational Planning of Bike-Sharing Systems by Data Mining - A Case Study,” in Computational Logistics.

[6] P. Borgnat, E. Fleury, C. Robardet, and A. Scherrer, “Spatial Analysis of Dynamic Movements, Lyon’s Shared Bicycle Program,”in European ConferenceonComplexSystems(ECCS).

[7] L. MetroBike, “2016 Year-end wrap-up will appear at the end of January,” http://bike-sharing.blogspot.com/2017/01/ 2016-year-end- wrap-up-will-appear-at.html.