

A Project Report on
Bike Share Prediction Project

Submitted in partial fulfillment of the requirements for
the award of the degree of

Master of Science in Data Science and Big Data Analytics

in

Data Science and Big Data Analytics

by

Dhaval Tharkar

41867

Under the Guidance of

Prof. Esmita Gupta



Department of Information Technology

B. K. Birla College of Arts, Science and Commerce (Autonomous), Kalyan

B. K. Birla College Road, Near RTO, Kalyan

UNIVERSITY OF MUMBAI

Academic Year 2021-2022

Acknowledgement

This Project Report entitled “*Bike Share Prediction Project*” Submitted by “*Dhaval Tharkar*” (41867) is approved for the partial fulfilment of the requirement for the award of the degree of *Master of Science* in **Data Science and Big Data Analytics** from *University of Mumbai*.

Co-Guide

Prof. Esmita Gupta
Guide

Prof. Esmita Gupta
Head, Department of Information Technology

Place: B. K. Birla College (Autonomous), Kalyan
Date:

CERTIFICATE

This is to certify that the project entitled ***“Bike Share Prediction Project”*** submitted by ***“Dhaval Tharkar” (41867)*** for the partial fulfilment of the requirement for award of a degree ***Master of Science in Data Science and Big Data Analytics***, to the University of Mumbai, is a bonafide work carried out during academic year 2021-2022.

Co-Guide

Prof. Esmita Gupta
Guide

Prof. Esmita Gupta
Head Department of IT

Dr. Avinash Patil
Principal

External Examiner(s)

1.

2.

Place: B. K. Birla College (Autonomous), Kalyan
Date

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

Dhaval Tharkar student Id: - 41867

Date:

Abstract

This dataset contains the hourly and daily count of rental bikes between years 2011 and 2012 in Capital bikeshare system with the corresponding weather and seasonal information.

This case study is about a bike rental shop. They want to predict the demand of bikes at any given hour of the day, so that, they can arrange for sufficient number of bikes for the customers.

Bike sharing systems are a new generation of traditional bike rentals where the whole process from membership, rental and return back has become automatic. Through these systems, users are able to easily rent a bike from a particular position and return back at another position. Currently, there are about over 500 bike-sharing programs around the world which is composed of over 500 thousand bicycles. Today, there exists great interest in these systems due to their important role in traffic, environmental and health issues. Apart from interesting real-world applications of bike sharing systems, the characteristics of data being generated by these systems make them attractive for the research.

The goal here is to build an end-to-end regression task. Here the user will provide the data and the result will be given by the best performing hyper tuned Machine Learning model. The user will also get privileges to choose the deployment options.

Table Content

	INDEX	Pg. No.	Sign
CHAPTER 1	INTRODUCTION	7	
1.1	Background	8	
1.2	Objective	8	
1.3	Purpose, Scope & Applicability	8	
1.3.1	Purpose	8	
1.3.2	Scope	9	
1.3.3	Applicability	9	
1.4	Achievements	10	
CHAPTER 2	SURVEY OF TECHNOLOGIES	11	
CHAPTER 3	REQUIREMENTS AND ANALYSIS	19	
3.1	Problem Definition	19	
3.2	Requirements Specification	19	
3.3	Planning and Scheduling	19	
3.4	Software and Hardware Requirements	20	
3.5	Preliminary Product Description	20	
3.6	Preparing a Model	21	
CHAPTER 4	METHODOLOGY	23	
CHAPTER 5	EVENT LIST	25	
CHAPTER 6	TABLE LIST	26	
CHAPTER 7	APPENDIX	27	
CHAPTER 8	SOURCE CODE	31	
CHAPTER 9	SCREENSHOTS	35	
CHAPTER 10	CONCLUSION	38	
CHAPTER 11	REFERENCE	39	

CHAPTER 1

1. INTRODUCTION

This project is aimed to developing a Bike Share Prediction Project. This is an easy to operate user friendly, economically priced software that meet the requirements of online bike share model to manage all the kind of booking of bike and can check for the availability of the bikes on to the website. Customer can predict the availability of the bikes on website by the help of prediction page. Bike Sharing model is getting increasing into our day-to-day life style. By using Bike sharing website for sharing the bike pollution is controlled and less amount of carbon footprint in produced.

Bike sharing systems are a new generation of traditional bike rentals where the whole process from membership, rental and return back has become automatic. Through these systems, users are able to easily rent a bike from a particular position and return back at another position. Currently, there are about over 500 bike-sharing programs around the world which is composed of over 500 thousand bicycles. Today, there exists great interest in these systems due to their important role in traffic, environmental and health issues. Apart from interesting real-world applications of bike sharing systems, the characteristics of data being generated by these systems make them attractive for the research.

The goal here is to build an end-to-end regression task. Here the user will provide the data and the result will be given by the best performing hyper tuned Machine Learning model.

The user will also get privileges to choose the deployment options.

1.1 Background:

Presently in Bike Share Prediction System, the application manages all the activities related to Booking and checking of the bikes availability onto the website automatically by taking some of the input from the user and tell them the availability of bikes. So, a registered users on website can check for the casual users. Registered users can register themselves and their bike for the further use, if any casual users are there who want the specific ride from one start location to their destination location.

1.2 OBJECTIVES

The objective of software project planning is to provide a framework that enables the Casual Users to make reasonable estimates of resources, cost, and schedule a share ride for themselves. These estimates are made within limited time frame at the beginning of a software project and should be updated regularly as the project progresses. In addition, estimates should attempt to define best case and worst-case scenarios so that project outcomes can be bounded.

The main objective of the project entire activity is to automate the sharing of the bikes process of day-to-day activities of system Library like:

1. register the Users as Registered user and their bikes.
2. Check whether the availability of the bike for casual users.
3. Re-locate the users destination location.
4. Cancellation of the bike availability.
5. Feedbacks Form.

1.3 Purpose, Scope & Applicability

1.3.1 Purpose

The proposed system focuses on easy and quick retrieval of data and minimization of errors. The proposed system is being developed using Python Programming Language in the IDLE as front end for the users and Database is been stored and accessed by MS Excel. Front end is windows-based application, which can interface with actual system. Back end is used to store and manipulate all the data, which is fed to the system.

Features of Proposed System The features of proposed system are described as follows: -

1. Automatic questionnaire.
2. Automatic data record maintain.
3. Easy retrieval of stored data.
4. Less time consuming.
5. Generation of bills.

6. Makes the entire process simpler.
7. More security is provided.
8. Quick access record.

1.3.2 Scope

The project has a very vast scope in future. The project can be implemented on internet in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. With the proposed software of Web Space Manager ready and fully functional the client is now able to manage and hence run the entire work in a much better, accurate and error free manner.

The following are the future scope for the project: -

The number of levels that the software is handling can be made unlimited in future from the current status of handling up to N levels as currently laid down by the software. Efficiency can be further enhanced and boosted up to a great extent by normalizing and de-normalizing the database tables used in the project as well as taking the kind of the alternative set of data structures and advanced calculation algorithms available.

We can in future generalize the application from its current customized status wherein other vendors developing and working on similar applications can utilize this software and make changes to it according to their business needs.

Faster processing of information as compared to the current system with high accuracy and reliability Automatic and error free report generation as per the specified format with ease.

Automatic calculation and generation of correct and precise Bills thus reducing much of the workload on the accounting staff and the errors arising due to manual calculations.

With a fully automated solution, lesser staff, better space utilization and peaceful work environment, the company is bound to experience high turnover.

A future application of this system lies in the fact that the proposed system would remain relevant in the future. In case there be any additions or deletion of the services, addition or deletion of any reseller in any type of modification in future can be implemented easily. The data collected by the system will be useful for some other purposes also. All these result in high client-satisfaction, hence, more and more business for the company that will scale the company business to new heights in the forthcoming future.

1.3.3 Applicability

The proposed system can be installed by the following users:

1. Users who want to register their bikes and use them as the sharing mode with others users.
2. All Users who want to ride a bike using sharing mode from one specific location to another.

1.3.4 Achievements

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configuration. It has got following feature.

1. Latest updates are made regularly on the site by administrator.
2. Ensure data accuracy.
3. The Changes made in the database are automatically reflected in our Site.
4. Feedback module is provided for all type of users.
5. Modification, deletion, adding of records becomes easier.
6. Process of data finding will take less time than paper work.

CHAPTER 2

SURVEY AND TECHNOLOGIES

INTRODUCTION TO PYTHON IDLE

IDLE stands for Integrated Development and Learning Environment. The story behind the name IDLE is similar to Python. Guido Van Rossum named Python after the British comedy group Monty Python while the name IDLE was chosen to pay tribute to Eric Idle, who was one of the Monty Python's founding members. IDLE comes bundled with the default implementation of the Python language since the 01.5.2b1 release. It is packaged as an optional part of the Python packaging with many Linux, Windows, and Mac distributions.

IDLE, as shown above, is a very simple and sophisticated IDE developed primarily for beginners, and because of its simplicity, it is highly considered and recommended for educational purposes. It offers a variety of features that you will look in detail along with examples later in this tutorial.

Some of the key features it offers are:

- Python shell with syntax highlighting,
- Multi-window text editor,
- Code autocompletion,
- Intelligent indenting,
- Program animation and stepping which allows one line of code to run at a time helpful for debugging,
- Persistent breakpoints,
- Finally, Call stack visibility.

How to Install IDLE?

- One way to install IDLE is to install Python from the official Python website, as shown below.

You can download the latest Python versions from this website for different operating systems like windows, Linux, and mac. It also provides you the docker images for Python, which you can directly use if you have a docker installed on your system.

INTRODUCTION TO JUPYTER NOTEBOOK

The Jupyter Notebook is an open-source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter.

Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

Getting Up and Running with Jupyter Notebook

The Jupyter Notebook is not included with Python, so if you want to try it out, you will need to install Jupyter.

There are many distributions of the Python language. This article will focus on just two of them for the purposes of installing Jupyter Notebook. The most popular is CPython, which is the reference version of Python that you can get from their website. It is also assumed that you are using Python 3.

Installation Guide:

If so, then you can use a handy tool that comes with Python called pip to install Jupyter Notebook like this:

```
$ pip install jupyter
```

The next most popular distribution of Python is Anaconda. Anaconda has its own installer tool called conda that you could use for installing a third-party package. However, Anaconda comes with many scientific libraries preinstalled, including the Jupyter Notebook, so you don't actually need to do anything other than install Anaconda itself.

Python Programming

Python is used for web development, AI, machine learning, operating systems, mobile application development, and video games. A successor to the ABC programming language, Python is a high level, dynamically typed language developed by Guido Van Rossum in the early 1980s.

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small- and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle-detecting garbage collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major revision that is not completely backward compatible with earlier versions. Python 2 was discontinued with version 2.7.18 in 2020.

Python consistently ranks as one of the most popular programming languages'

History of Python

Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC programming language, which was inspired by SETL capable of exception handling and interfacing with the Amoeba operating system.[10] Its implementation began in December 1989.[40] Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's "benevolent dictator for life", a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker. In January 2019, active Python core developers elected a five-member Steering Council to lead the project.

Python 2.0 was released on 16 October 2000, with many major new features. Python 3.0, released on 3 December 2008, with many of its major features backported to Python 2.6.x and 2.7.x. Releases of Python 3 include the 2to3 utility, which automates the translation of Python 2 code to Python 3.

Python 2.7's end-of-life was initially set for 2015, then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3. No further security patches or other improvements will be released for it. With Python 2's end-of-life, only Python 3.6.x and later are supported.

Python 3.9.2 and 3.8.8 were expedited as all versions of Python (including 2.7) had security issues leading to possible remote code execution and web cache poisoning.

Python Library

- Object-oriented signals that JavaScript's power to exert control over an HTML page is based on manipulating objects within that page.
- If you are familiar with object-oriented programming, you will be aware of some of the power that this can bring to the coding environment.

Numpy Library

NumPy (pronounced /'nʌmpaɪ/ (NUM-py) or sometimes /'nʌmpi/ (NUM-pee)) is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors. NumPy is a NumFOCUS fiscally sponsored project.

Pandas Library

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for

manipulating numerical tables and time series. It is free software released under the three-clause BSD license.

The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Its name is a play on the phrase "Python data analysis" itself. Wes McKinney started building what would become pandas at AQR Capital while he was a researcher there from 2007 to 2010

Matplotlib Library

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. SciPy makes use of Matplotlib.

Matplotlib was originally written by John D. Hunter. Since then it has an active development community and is distributed under a BSD-style license. Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in August 2012 and was further joined by Thomas Caswell. Matplotlib is a NumFOCUS fiscally sponsored project.

Matplotlib 2.0.x supports Python versions 2.7 through 3.10. Python 3 support started with Matplotlib 1.2. Matplotlib 1.4 is the last version to support Python 2.6. Matplotlib has pledged not to support Python 2 past 2020 by signing the Python 3 Statement.

Seaborn Library

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

For a brief introduction to the ideas behind the library, you can read the introductory notes or the paper. Visit the installation page to see how you can download the package and get started with it. You can browse the example gallery to see some of the things that you can do with seaborn, and then check out the tutorial or API reference to find out how.

To see the code or report a bug, please visit the GitHub repository. General support questions are most at home on stack overflow or discourse, which have dedicated channels for seaborn.

Spyder

Spyder is an open-source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates with a number of prominent packages in the scientific Python stack, including NumPy, SciPy, Matplotlib, pandas, IPython, SymPy and Cython, as well as other open-source software. It is released under the MIT license.

Initially created and developed by Pierre Raybaut in 2009, since 2012 Spyder has been maintained and continuously improved by a team of scientific Python developers and the community.

Spyder is extensible with first-party and third-party plugins, includes support for interactive tools for data inspection and embeds Python-specific code quality assurance and introspection instruments, such as Pyflakes, Pylint and Rope. It is available cross-platform through Anaconda, on Windows, on macOS through MacPorts, and on major Linux distributions such as Arch Linux, Debian, Fedora, Gentoo Linux, openSUSE and Ubuntu.

Spyder uses Qt for its GUI and is designed to use either of the PyQt or PySide Python bindings. QtPy, a thin abstraction layer developed by the Spyder project and later adopted by multiple other packages, provides the flexibility to use either backend.

HTML (HYPERTEXT MARKUP LANGUAGE)

The HyperText Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as `` and `<input />` directly introduce content into the page. Other tags such as `<p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997. A form of HTML, known as HTML5, is used to display video and audio, primarily using the `<canvas>` element, in collaboration with javascript.

CSS (Cascading Style Sheets)

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL.

Microsoft Excel

Microsoft Excel is a spreadsheet program included in the Microsoft Office suite of applications. ... Spreadsheets present tables of values arranged in rows and columns that can be manipulated mathematically using both basic and complex arithmetic operations and functions Relational Database.

What can you do with Microsoft Excel?

- Import, export, and convert Excel, .csv, .txt and .ods formatted data.
- Use editing formula to perform calculations on your data, and use formatting to create your own style.
- Chat in real time with others who are editing your spreadsheet while using the online version.
- Create charts with your data.

- Embed a spreadsheet — or individual sheets of your spreadsheet — on your blog or website.

Creating a Flask Application:

There are many modules or frameworks which allow building your webpage using python like a bottle, Django, Flask, etc. But the real popular ones are Flask and Django. Django is easy to use as compared to Flask but Flask provides you with the versatility to program with.

To understand what Flask is you have to understand a few general terms.

1. WSGI: Web Server Gateway Interface (WSGI) has been adopted as a standard for Python web application development. WSGI is a specification for a universal interface between the web server and the web applications.
2. Werkzeug: It is a WSGI toolkit, which implements requests, response objects, and other utility functions. This enables building a web framework on top of it. The Flask framework uses Werkzeug as one of its bases.
3. jinja2: jinja2 is a popular templating engine for Python. A web templating system combines a template with a certain data source to render dynamic web pages.

Flask is a web application framework written in Python. Flask is based on the Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

Installation:

We will require two packages to set up your environment. virtualenv for a user to create multiple Python environments side-by-side. Thereby, it can avoid compatibility issues between the different versions of the libraries and the next will be Flask itself.

virtualenv

pip install virtualenv

Create Python virtual environment

virtualenv venv

Activate virtual environment

windows > venv\Scripts\activate

linux > source ./venv/bin/activate

Flask Command: pip install Flask

Deploying Tool:

Heroku

Heroku is a cloud platform as a service (PaaS) supporting several programming languages. One of the first cloud platforms, Heroku has been in development since June 2007, when it supported only the Ruby programming language, but now supports Java, Node.js, Scala, Clojure, Python, PHP, and Go. For this reason, Heroku is said to be a polyglot platform as it has features for a developer to build, run and scale applications in a similar manner across most languages. Heroku was acquired by Salesforce in 2010 for \$212 million.

History of Heroku

Heroku was initially developed by James Lindenbaum, Adam Wiggins, and Orion Henry for supporting projects that were compatible with the Ruby programming platform known as Rack. The prototype development took around six months. Later on, Heroku faced setbacks because of lack of proper market customers as many app developers used their own tools and environment. citation needed] In January 2009, a new platform was launched which was built almost from scratch after a three-month effort. In October 2009, Byron Sebastian joined Heroku as CEO. On December 8, 2010, Salesforce.com acquired Heroku as a wholly owned subsidiary of Salesforce.com. On July 12, 2011, Yukihiro "Matz" Matsumoto, the chief designer of the Ruby programming language, joined the company as Chief Architect, Ruby. That same month, Heroku added support for Node.js and Clojure. On September 15, 2011, Heroku and Facebook introduced Heroku for Facebook. At present Heroku supports Redis databases in addition to its standard PostgreSQL.

Etymology

The name "Heroku" is a portmanteau of "heroic" and "haiku". The Japanese theme is a nod to Matz for creating Ruby. The name itself is pronounced similarly to the Japanese word meaning "widely" (heroku), though the creators of Heroku did not want the name of their project to have a particular meaning, in Japanese or any other language, and so chose to invent a name.

Heroku Architecture

Applications that are run on Heroku typically have a unique domain used to route HTTP requests to the correct application container or dyno. Each of the dynos are spread across a "dyno grid" which consists of several servers. Heroku's Git server handles application repository pushes from permitted users.

All Heroku services are hosted on Amazon's EC2 cloud-computing platform.

CHAPTER 3

3. REQUIREMENT AND ANALYSIS

3.1 PROBLEM DEFINITION

In the current system, the admin keeps the record of Bike Sharing Model System. Also, the admin himself has to check-out for the availability bikes for Casual Users to use the sharing bike for their need. Keeping checking on the registered Users details was a tedious task. Also, present system is complicated for maintaining data.

3.2 REQUIREMENTS SPECIFICATION

Our system can be implemented in a wide range of hardware, but it has to meet some performance requirements. As a portable unit car tooth should be small and light. Runs on batteries therefore low power consumption is very important. Moreover, its CPU must be able to co-operate with many internal modules.

Outside unit is required to have enough computational power to server CTs in a short time. Since establishing Bluetooth connection takes a few seconds, other operations, such as time-consuming authorization or making a transaction, should be done as fast as possible. On the other hand, the multifunction Administration Center needs computational resources to process a large amount of data. It is also necessary to ensure an appropriate level of transaction security without lowering the speed of communication.

3.3 PLANNING AND SCHEDULING

PLANNING: Project planning defines the project activities and end products that will be performed and describes how the activities will be accomplished. The purpose of project planning is to define each major task, estimate the time and resources required, and provide a framework for management review and control. The project planning activities and goals include defining:

- The specific work to be performed and goals that define and bind the project.
- Estimates to be documented for planning, tracking, and controlling the project
- Commitments that are planned, documented, and agreed to by affected groups.
- Project alternatives, assumptions, and constraint

SCHEDULING: The project schedule provides a graphical representation of predicted tasks, milestones, dependencies, resource requirements, task duration, and deadlines. The project's master schedule interrelates all tasks on a common time scale. The project schedule should be detailed enough to show each WB task to be performed, the name of the person responsible for completing the task, the start and end date of each task, and the expected duration of the task.

- Define the type of schedule
- Define precise and measurable milestones

- Estimate task duration
- Define priorities
- Define the critical path
- Document assumptions
- Identify risks

3.4 SOFTWARE AND HARDWARE REQUIREMENTS

Software: -

- Front End: Python (IDLE), Jupyter Notebook, Spyder.
- Back End: Microsoft Excel.
- Operating System: Windows 10

Hardware: -

- Processor: Intel Core i5
- RAM: 8GB
- Hard Disk: 1TB
- Monitor: Monitor :15" CRT

3.5 PRELIMINARY PRODUCT DESCRIPTION

The first step in the system development life cycle in the preliminary investigation to determine the feasibility of the system. The purpose of the preliminary investigation is to evaluate project requests. It is not a design study nor does it include the collection of details to describe the business system in all respect. Rather, it is the collecting of information that helps committee members to evaluate the merits of the project request and make an informed judgment about the feasibility of the proposed project.

Analysts working on the preliminary investigation should accomplish the following objectives:

- Clarify and understand the project request.
- Determine the size of the project.
- Assess costs and benefits of alternative approaches.
- Determine the technical and operational feasibility of alternative approaches.
- Report the findings to management, with recommendations outlining the Acceptance or rejection of the proposal.
- Benefit to Organization.

- The Organization will obviously be able to gain benefits such as savings in operating cost reduction in paperwork, better utilization of human resources and more presentable image increasing goodwill

The Initial Cost:

The initial cost of setting up the system will include the cost hardware software (OS add –on software).

3.6 PREPARING A MODEL

Linear Regression Model

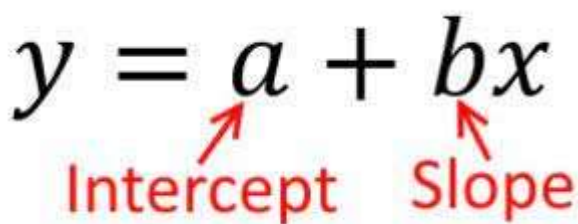
Linear regression is used to predict the relationship between two variables by applying a linear equation to observed data. There are two types of variable, one variable is called an independent variable, and the other is a dependent variable. Linear regression is commonly used for predictive analysis. The main idea of regression is to examine two things. First, does a set of predictor variables do a good job in predicting an outcome (dependent) variable? The second thing is which variables are significant predictors of the outcome variable? In this article, we will discuss the concept of the Linear Regression Equation, formula and Properties of Linear Regression.

Linear Regression Formula

The measure of the relationship between two variables is shown by the correlation coefficient. The range of the coefficient lies between -1 to +1. This coefficient shows the strength of the association of the observed data between two variables.

Linear Regression Equation is given below:

$$Y=a+bX$$



The diagram shows the equation $y = a + bx$ in a large, bold, black font. Below the letter 'a' is the word "Intercept" in red, with a red arrow pointing from the word to 'a'. Below the letters 'bx' is the word "Slope" in red, with a red arrow pointing from the word to 'b'.

where X is the independent variable and it is plotted along the x-axis

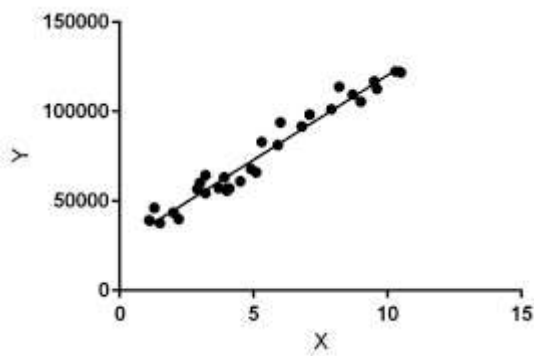
Y is the dependent variable and it is plotted along the y-axis

Here, the slope of the line is b, and a is the intercept (the value of y when x = 0).

Linear Regression Formula

As we know, linear regression shows the linear relationship between two variables. The equation of linear regression is similar to that of the slope formula. We have learned this formula before in earlier classes such as a linear equation in two variables. Linear Regression Formula is given by the equation

$$Y = a + bX$$



CHAPTER 4

METHODOLOGY

Proposed Model: -



- **Data collection:**

Solving machine learning problems firstly we require raw data because without raw data we cannot do machine learning problems. raw data we get from further discussion of the problem with client and data scientist team we focus on data that is a data integration and data integration is a very difficult task because we collect data from multiple resources like structure data unstructured data, web scraping, etc. collected data stored in data warehouse and we get data from a data warehouse.

- **Data Pre-processing:**

Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data pre-processing task.

- **Data splitting:**

The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model. In this step data is split for training and testing almost 80% of data is for training and 20% for testing is a basic rule in the machine learning.

- **Training:**

In this step, we do training data for machine analysis itself and we do another step is to validate training data because training data set will produce either overfitting or under the fitting problem that means false positive output or true negative output that means overfitting means when you go new area and 1st person give disrespect and you considering all people are same this is.

- **Testing and Evaluation:**

In the testing phase, we test the model using cross-validation, we check the model is well or not and going is right or not, there are some techniques of cross-validation and we use confusion matrix for checking model performance. We will test in all algorithms and will get best suitable output.

CHAPTER 5

Event List

Sr No	Form Name	Task
1.	Home Page	To display the Home Page
2.	About Page	To display the About Page
3.	Prediction Page	To Predict the Prediction values
4.	Contact Page	To display the Contact Page
5.	app.py	To make the application
6.	model.py	To make the model

CHAPTER 6

TABLE LIST

hours.csv

Column Name	Data Types
instant	int64
dteday	Object
season	int64
yr	int64
mnth	int64
hr	int64
holiday	int64
weekday	int64
workingday	int64
weathersit	int64
temp	float64
atemp	float64
hum	float64
windspeed	float64
Casual	int64
Registered	int64
Cnt	int64

CHAPTER 7

Appendix

Installation of Anaconda

1. Go to the Anaconda Website and choose a Python 3.x graphical installer (A) or a Python 2.x graphical installer (B). If you aren't sure which Python version you want to install, choose Python 3. Do not choose both.
2. Locate your download and double click it. ... When the screen below appears, click on Next.
3. Read the license agreement and click on I Agree.
4. Click on Next.
5. Note your installation location and then click Next.
6. This is an important part of the installation process. The recommended approach is to not check the box to add Anaconda to your path. This means you will have to use Anaconda Navigator or the Anaconda Command Prompt (located in the Start Menu under "Anaconda") when you wish to use Anaconda (you can always add Anaconda to your PATH later if you don't check the box). If you want to be able to use Anaconda in your command prompt (or git bash, cmd, powershell etc), please use the alternative approach and check the box.
7. Click on Next.
8. You can install Microsoft VSCode if you wish, but it is optional.
9. Click on Finish.

Creating an environment onto the Anaconda Navigator using Anaconda Prompt

Open Anaconda prompt and create new environment

```
conda create -n your_env_name python = (any_version_number)
```

Then Activate the newly created environment

```
conda activate your_env_name
```

Clone the repository using git

```
git clone github_link
```

Change to the cloned directory

```
cd <directory_name>
```

To install all requirement packages for the app

```
pip install -r requirements.txt
```

Then, Run the app

Web: gunicorn main:app --reload

Install necessary libraries

Numpy

```
conda install numpy
```

Pandas

```
conda install pandas
```

Matplotlib

```
conda install matplotlib
```

Seaborn

```
conda install seaborn
```

Sklearn

```
conda install scikit-learn
```

Scipy

```
conda install -c anaconda scipy
```

Klib

```
conda install klib
```

Flask

```
conda install flask
```

How to open the application into the local Desktop: -

Before opening the file first save all the file into a separate folder of our local pc Desktop.

Within the main project folder create a folder named as static and within that folder create a folder named as css and then save all necessary css file into it.

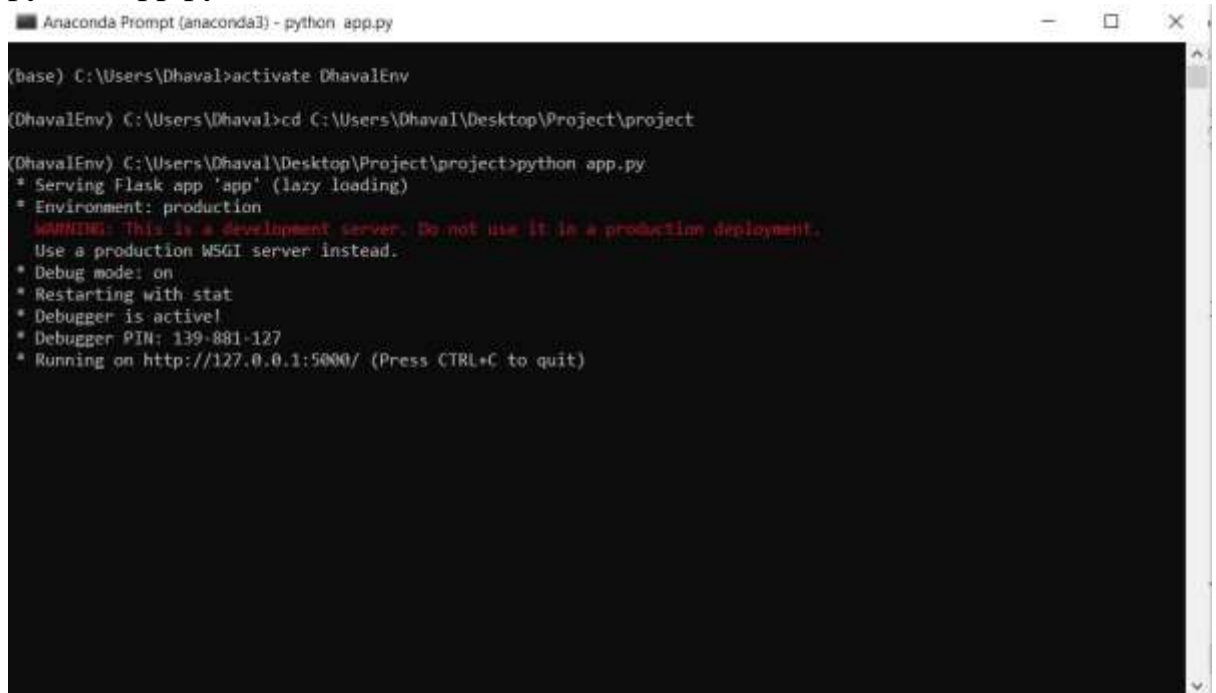
Within the main project folder create a folder named as templates and save the all-html file into it.

Save model.pkl and app.py file into the main project folder.

Steps to open:

1. Open anaconda prompt.

2. Activate the environment using the command:
activate enviroment_name
3. Change the directory to the main project directory using the command:
cd path of the folder
4. To open the application, use the below command:
python app.py



```
Anaconda Prompt (anaconda3) - python app.py

(base) C:\Users\Dhaval>activate DhavalEnv

(DhavalEnv) C:\Users\Dhaval>cd C:\Users\Dhaval\Desktop\Project\project

(DhavalEnv) C:\Users\Dhaval\Desktop\Project\project>python app.py
* Serving Flask app 'app' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 139-881-127
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

To open the application on to the Local Desktop, copy the link of “ <http://127.0.0.1.5000>” and paste it to the browser to access it.

How to deploy application on cloud.

Before deploying we will use Heroku, before creating an application of our project upload all the necessary folder and files into the GitHub Repository.

GitHub Repository Links

<https://github.com/Dhavaltharkar/bike-share-prediction>

1. Create an account on Heroku using the same email used for GitHub and then sign to it.
2. Create an app using create app button give the unique name to it
3. Link the GitHub account with the Heroku.
4. And then click on Deploy Branch.
5. After deploying click on view to View our application
6. <https://bikesharepredictiondemo.herokuapp.com/>

If there are some errors, we can check those errors by using Heroku CLI

1. Login to the Heroku CLI
2. Write a command of Heroku CLI
`heroku logs -tail -app <app_name>`

CHAPTER 8

SOURCE CODE

model.py

```
# Importing the libraries

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import pickle


#import the csv file
bike_df=pd.read_csv("hour.csv")


#Rename the columns
bike_df.rename(columns={'instant':'rec_id','dteday':'datetime','yr':'year','mnth':'month','weather':'weather_condition',
                        'hum':'humidity','cnt':'total_count'},inplace=True)


# Changing the dtype of necessary columns to categorical columns
bike_df['season']=bike_df.season.astype('category')
bike_df['year']=bike_df.year.astype('category')
bike_df['month']=bike_df.month.astype('category')
bike_df['holiday']=bike_df.holiday.astype('category')
bike_df['weekday']=bike_df.weekday.astype('category')
bike_df['workingday']=bike_df.workingday.astype('category')
bike_df['weather_condition']=bike_df.weather_condition.astype('category')


# Dropping Unnecessary Columns
bike_df.drop('datetime',axis=1,inplace=True)
bike_df.drop('rec_id',axis=1,inplace=True)
bike_df.drop('year',axis=1,inplace=True)
```

```
#load the required libraries

from sklearn import preprocessing,metrics,linear_model
from sklearn.model_selection import cross_val_score,cross_val_predict,train_test_split


# Splitting the X and y Values from the dataset
# X is an Independent Variable
X = bike_df.iloc[:, :13]


# y is a dependent variables
y = bike_df.iloc[:, -1]


from sklearn.linear_model import LinearRegression
regressor = LinearRegression()


#Fitting model with trainig data
regressor.fit(X, y)


# Saving model to disk
pickle.dump(regressor, open('model.pkl','wb'))
```


app.py

```
from flask import Flask, request, jsonify, render_template
import pickle

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('home.html')

@app.route('/about')
def about():
    return render_template('about.html')

@app.route('/contact')
def contact():
    return render_template('contact.html')

@app.route('/predict', methods = ['GET', 'POST'])
def predict():
    if request.method == 'POST':
        season = int(request.form['season'])
        month = int(request.form['month'])
        hr = int(request.form['hr'])
        holiday = int(request.form['holiday'])
        weekday = int(request.form['weekday'])
        workingday = int(request.form['workingday'])
        wheather_condition = float(request.form.get('wheather_condition', False))
        temp = int(request.form['temp'])
```

```

    atemp = int(request.form['atemp'])
    humidity = int(request.form['humidity'])
    windspeed = int(request.form['windspeed'])
    casual = int(request.form['casual'])
    registered = int(request.form['registered'])

    prediction =
model.predict([[season,month,hr,holiday,weekday,workingday,wheather_condition,temp,atemp,humidity,windspeed,casual,registered]])

    output=round(prediction[0])

    if output<=0:
        return render_template('prediction.html',prediction_texts="Sorry bike not available")
    else:
        return render_template('prediction.html',prediction_text="Number of Bike available
for sharing : { }".format(output))
    else:
        return render_template('prediction.html')

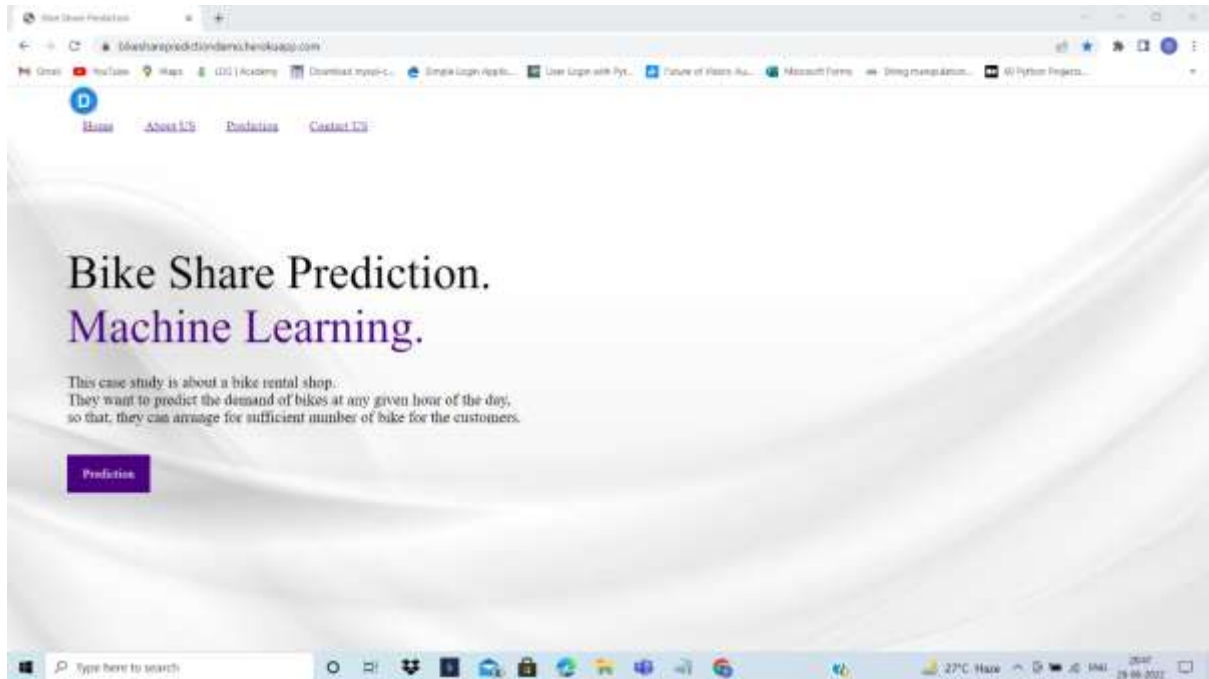
if __name__ == "__main__":
    app.run(debug=True)

```

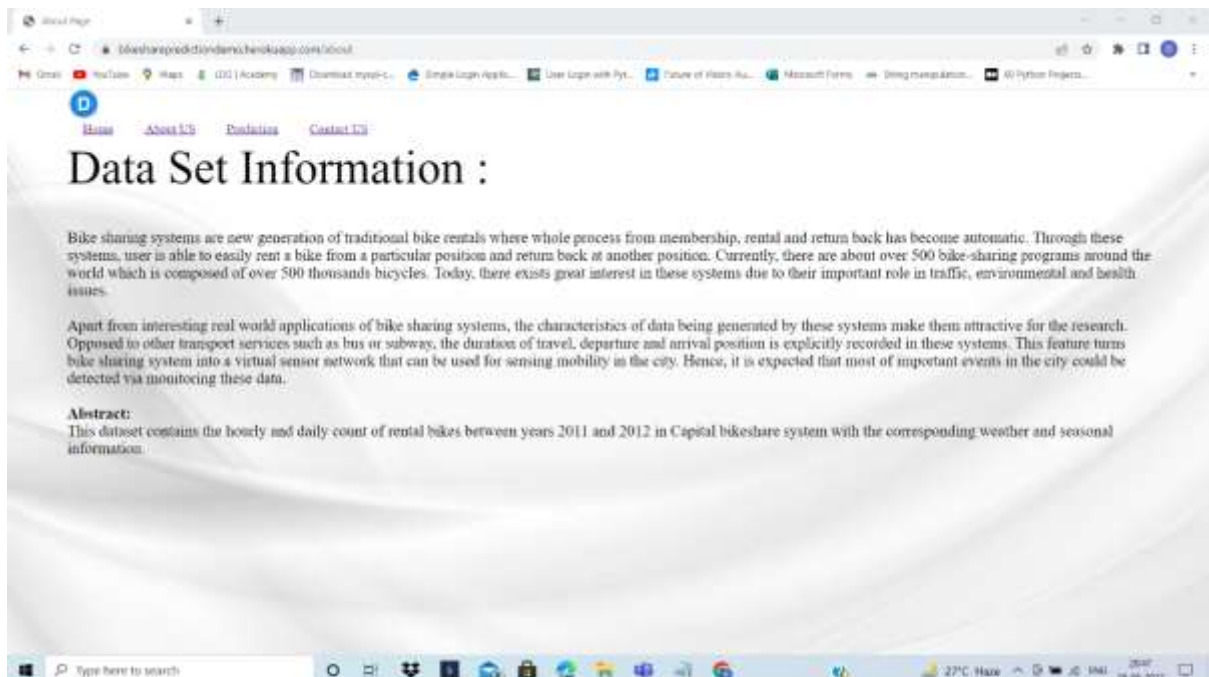
CHAPTER 9

SCREENSHOTS

home.html



about.html



prediction.html

The screenshot displays a web browser window with the URL `bluesharpredictionsdemo.herokuapp.com/predict`. The page title is "Bike Share Prediction Project" with the instruction "Fill the form for prediction".

At the top left is a blue "Back" button. Below it are several dropdown menus for selecting variables:

- season: -- select season --
- Month: -- select month --
- Hour: -- select hour --
- Holiday: -- select Holiday --
- Weekday: -- select Weekday --
- Workingday: -- select Workingday --

Below these are text input fields for the following variables:

- weather_condition: -- select weather_condition --
- Enter Temperature in Celsius: [empty field]
- Enter Temperature Feeling in Celsius: [empty field]
- Enter Humidity: [empty field]
- Enter Wind_Speed: [empty field]
- Enter Casual Users Id: [empty field]
- Enter Registered Users Id: [empty field]

At the bottom left is a blue "Predict" button. The browser's taskbar at the bottom shows the system clock as 25:49 on 29-05-2022 and the weather as 27°C Haze.

contact.html

The image shows a web browser window displaying a contact form. The browser's address bar shows the URL "bbsksharepredictorindemo.herokuapp.com/contact". The page has a blue header with a "Back" button. The main heading is "Contact to Online Bike Share Service Provider Center". Below the heading, there are five input fields: "Enter Full Name:", "Enter Subject:", "Enter Phone Number:", "Enter Email Id:", and "Enter Message:". The "Enter Message:" field is a larger text area. At the bottom of the form, there is a red button labeled "Send Contact Details". The browser's taskbar at the bottom shows various application icons and the system clock indicating 25/04/2022 at 20:48.

Back

Contact to Online Bike Share Service Provider Center

Enter Full Name:

Enter Subject:

Enter Phone Number:

Enter Email Id:

Enter Message:

Send Contact Details

CHAPTER 10

CONCLUSION

The model which we were making is to predict a value of number of bikes availability using machine learning techniques. We have collected the data of Bike Share from UCI having attributes like different columns like their year, season, month, hr, holiday, weekday, working day, etc. The data is then processed using different algorithms where results of each algorithm is compared, getting Linear Regression algorithm as the most accurate amongst them, so we have used Linear Regression model because irrespective of size it runs efficiently and gives more accuracy than any other algorithm. Further it would be available in GUI as a Web-application developed using Python flask making it user friendly so that users could give input and get the availability of bike numbers according to it.

CHAPTER 11

REFERENCE

- <https://towardsdatascience.com/>
- <https://geeksforgeeks.org/>
- <https://github.com/>
- <https://www.kaggle.com/>
- <https://w3schools.com/python/>
- Data Analysis with Microsoft Excel (Author: Kenneth N. Berk, Patrick Carey).
- Data Analytics: Mode Accessible (Dr. Anil Maheshwari).
- Learning Python, O'REILLY PUBLICATIONS (Author: - Mark Lutz).
- Head First Python, O'REILLY PUBLICATIONS (Author: - Paul Barry).
- Python Data Science Handbook, O'REILLY PUBLICATIONS (Author: -Jake VanerPlas).
- Python for Unix and Linux System Administration, O'REILLY PUBLICATIONS (Author: - Noah Gift & Jeremy M. Jones).
- Purnima Sachdeva and K N Sarvanan, Prediction of Bike Sharing Demand, Oriental Journal of Computer Science and Technology, Article Published: 21 Mar 2017
- Shaheen, Susan, Stacey Guzman, and Hua Zhang. "Bikesharing in Europe, the Americas, and Asia: past, present, and future." Transportation Research Record: Journal of the Transportation Research Board 2143 (2010).
- Daniele Gammelli, Yihua Wang, Dennis Prak, Francisco Camara Pereira, Predictive and Prescriptive Performance of Bike-Sharing Demand Forecasts for Inventory Management, Research Gate (July2021).