A REPORT ON

"*BIKE SHARING*"

Submitted for partial fulfilment of award of

DEGREE
OF
BACHELOR OF COMPUTER APPLICATIONS

Submitted By –
HEMANT GUPTA

Roll No: 210934106126

Under the supervision of
Prateek Gupta

INSTITUTE OF TECHNOLOGY & SCIENCE

MOHAN NAGAR, GHAZIABAD

Batch: 2021-2024

Acknowledgement

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them.

I am highly indebted to Mr. Prateek Gupta for their guidance and supervision as well as for providing necessary information regarding the project and also for their support in completing the project. His constant guidance and willingness to share his vast knowledge made me understand this project and its manifestations in great depths and helped me to complete the assigned tasks on time.

Hemant Gupta
210934106126

CERTIFICATE

This is to Certify that **HEMANT GUPTA** has carried out the project work presented in this report entitled **"BIKE SHARING"** for the award of **Bachelor Of Computer Applications** from Institute of Technology & Science, Mohan Nagar, Ghaziabad, under my supervision. The report embodies result of original work and studies carried out by Student himself and the contents of the report do not form the basis for the award of any other degree to the candidate or to anybody else.

Date:11-12-2023

PRATEEK GUPTA

DATA SCIENTIST

TRAINER

INDEX

CHAPTER-1

INTRODUCTION

Bike sharing systems are a means of renting bicycles where the process of obtaining membership, rental, and bike return is automated via a network of kiosk locations throughout a city. Using these systems, people are able rent a bike from a one location and return it to a different place on an as-needed basis. Currently, there are over 500 bike-sharing programs around the world. Several bike/scooter rides sharing facilities (e.g., Bird, Capital Bikeshare, Citi Bike) have started up lately especially in metropolitan cities like San Francisco, New York, Chicago and Los Angeles, and one of the most important problems from a business point of view is to predict the bike demand on any particular day. While having excess bikes results in wastage of resource (both with respect to bike maintenance and the land/bike stand required for parking and security), having fewer bikes leads to revenue loss (ranging from a short-term loss due to missing out on immediate customers to potential longer-term loss due to loss in future customer base), Thus, having an estimate on the demands would enable efficient functioning of these companies. The goal of this project is to combine the historical bike usage patterns with the weather data to forecast bike rental demand. The data set consists of hourly rental data spanning two years. The training set is comprised of the first 19 days of each month, while the test set is the 20th to the end of the month.

1.1 Overview and Problem Statement

Overview

Urban transportation faces ongoing challenges, including traffic congestion, environmental pollution, and the need for sustainable mobility solutions. In response to these issues, the bike-sharing initiative is introduced as an innovative and eco-friendly approach to urban commuting. This initiative provides individuals with a convenient, accessible, and healthy alternative to traditional transportation methods.

Problem Statement

- Traffic Congestion:

Urban areas are grappling with increasing traffic congestion, leading to longer commute times and decreased overall efficiency. Traditional modes of transportation contribute significantly to this issue, necessitating the exploration of alternative options.

- Environmental Impact:

Conventional transportation methods heavily rely on fossil fuels, contributing to environmental pollution and climate change. The need for eco-friendly

alternatives is more critical than ever to mitigate the adverse impact on the environment.

- Limited Sustainable Mobility:

Many urban residents face challenges in accessing sustainable transportation options. Public transportation may not be readily available, and private vehicle ownership can be impractical or financially burdensome. This gap in mobility solutions calls for an accessible and cost-effective alternative.

Health and Lifestyle:

Sedentary lifestyles associated with conventional commuting methods contribute to various health issues. The lack of physical activity in daily routines has prompted the need for solutions that not only address transportation challenges but also promote a healthier lifestyle.

1.2 Purpose

1. Sustainable Transportation:

- Reducing Carbon Emissions: Bike sharing promotes a sustainable mode of transportation that produces minimal carbon emissions, contributing to environmental conservation and combating climate change.
- Energy Efficiency: Bicycles are energy-efficient, requiring less energy for transportation compared to motorized vehicles.

2. Urban Mobility Enhancement:

- Reducing Traffic Congestion: Bike sharing helps alleviate traffic congestion in urban areas by providing an alternative mode of transportation for short-distance trips.

- Last-Mile Connectivity: Addresses the "last-mile" problem by providing a convenient solution for commuters to reach their final destination from public transit hubs.

3. Public Health Promotion:
- Physical Activity: Encourages physical activity and an active lifestyle, contributing to improved public health by reducing sedentary behavior associated with traditional commuting methods.
- Reducing Air Pollution: The shift towards cycling helps reduce air pollution, benefiting the respiratory health of individuals and the community.

4. Cost-Effective Transportation:
- Affordability: Bike sharing offers a cost-effective alternative for short-distance travel, often more affordable than public transportation or private vehicle ownership.
- Economic Savings: Reduces the economic burden associated with maintaining and operating private vehicles.

5. Enhanced Accessibility:
- Access for All: Increases accessibility to transportation, especially in areas with limited public transit options, promoting inclusivity and equal access to mobility solutions.

1.3 Scope

- Geographical Coverage:

Define the specific areas or cities where the bike-sharing service will be implemented. This could range from a small neighborhood to an entire metropolitan area.

- Scale of Operation:

Determine the size and scale of the bike-sharing program, including the number of bikes, docking stations, and operational zones.

- Target Audience:

Identify the demographic groups that the bike-sharing service aims to serve. This could include commuters, students, tourists, or residents of specific neighborhoods.

- Integration with Public Transit:

Explore opportunities for integration with existing public transportation systems. This could involve placing bike-sharing stations near bus stops or train stations to facilitate seamless transfers.

- Pricing Structure:

Define the pricing model for bike rentals. Consider whether it will be a subscription-based model, pay-as-you-go, or a combination of both. Determine any additional fees, discounts, or incentives for users.

1.4 Tools Used

- Python Programming Language:

Python is chosen as the primary programming language due to its versatility, extensive libraries, and widespread use in data science and machine learning.

- NumPy and Pandas:

NumPy and Pandas are employed for data manipulation and preprocessing tasks. NumPy facilitates numerical operations, while Pandas is used for structured data manipulation and analysis.

- Scikit-Learn:

Scikit-Learn, a powerful machine learning library in Python, is employed for the implementation of the K-Means clustering algorithm. It provides user-friendly tools for data analysis and modeling.

- Matplotlib and Seaborn:

Matplotlib and Seaborn are used for data visualization. These libraries enable the creation of insightful plots and charts to visually represent patterns and clusters within the customer data.

- Jupyter Notebooks:

Jupyter Notebooks serve as an interactive computing environment, allowing for the development, execution, and documentation of code in a collaborative and transparent manner.

CHAPTER-2

System Analysis

2.1 Identification Of Need

- Traffic Congestion:

Evaluate the level of traffic congestion in urban areas. If there is a significant problem with traffic jams and congestion, a bike-sharing program can offer an alternative mode of transportation, especially for short-distance trips.

- Environmental Concerns:

Assess the environmental impact of traditional transportation methods, such as carbon emissions from vehicles. If there is a strong emphasis on reducing environmental pollution and promoting sustainable practices, bike sharing becomes a viable solution.

- Health and Lifestyle:

Consider the health implications of sedentary lifestyles and the lack of physical activity associated with traditional commuting. If there is a growing awareness and emphasis on promoting an active lifestyle, bike sharing can encourage physical activity.

- Tourism and Local Attractions:

Consider the presence of tourist destinations, cultural sites, or local attractions. Bike sharing can be an attractive and efficient way for tourists and locals alike to explore and navigate such areas.

- Cost-Effective Solutions:

Evaluate the economic feasibility of bike sharing compared to other transportation options. If there is a need for cost-effective alternatives for short-distance travel, bike sharing can offer an affordable solution.

2.2 Preliminary Investigation

Preliminary investigation in the context of bike sharing involves conducting a thorough assessment and analysis to gather relevant information and insights before the actual implementation of the bike-sharing program. Here are key steps and considerations for a preliminary investigation in bike sharing:

- Stakeholder Identification:

Identify and engage with key stakeholders, including local government agencies, urban planners, potential users, businesses, and community organizations. Understand their perspectives, concerns, and potential support for a bike-sharing initiative.

- Market Research:

Conduct market research to assess the demand for bike sharing in the target area. Understand the demographics, commuting patterns, and preferences of the potential user base. Evaluate existing transportation infrastructure and services.

- Legal and Regulatory Compliance:

Investigate and understand local laws, regulations, and permit requirements related to bike-sharing services. Ensure compliance with zoning regulations, traffic laws, and any other legal considerations.

- Environmental Impact Assessment:

Evaluate the potential environmental impact of introducing a bike-sharing program. Consider factors such as reduced carbon emissions, improved air quality, and overall sustainability.

- Infrastructure Assessment:

Assess the existing infrastructure, including roads, bike lanes, and public spaces. Determine the feasibility of integrating bike-sharing stations into the urban landscape and identify areas for potential expansion.

2.3 Feasibility Study

A feasibility study for a bike-sharing program involves a comprehensive analysis of various aspects to determine whether the initiative is viable and likely to succeed. Here are key components to include in a feasibility study for bike sharing:

1. Market Feasibility:

Demand Analysis: Assess the demand for bike-sharing services in the target area. Analyze demographic data, commuting patterns, and factors influencing the need for alternative transportation.

Competitor Analysis: Evaluate existing transportation options, including bike rentals, public transit, and private vehicles. Identify potential competitors and understand their strengths and weaknesses.

User Preferences: Conduct surveys or focus groups to understand user preferences, expectations, and potential obstacles to bike-sharing adoption.

2. Technical Feasibility:

Infrastructure Assessment: Evaluate the adequacy of existing infrastructure, including roads, bike lanes, and public spaces, to support bike-sharing stations.

Technology Requirements: Assess the availability and reliability of technologies such as GPS tracking, mobile apps, payment gateways, and smart locks. Ensure the technical feasibility of integrating these components into the bike-sharing system.

Operational Scalability: Evaluate whether the chosen technology can scale to meet the potential demand for the bike-sharing program.

3. Financial Feasibility:

Cost Estimation: Estimate the initial setup costs, including bike acquisition, station installation, technology implementation, and marketing expenses.

Operational Costs: Project ongoing operational costs, such as maintenance, repairs, marketing, and staff salaries.

Revenue Projections: Develop revenue projections based on user fees, partnerships, sponsorships, or other potential income streams.

Return on Investment (ROI): Calculate the anticipated ROI and payback period for the bike-sharing program.

CHAPTER-3

Means Of Project

3.1 Hardware Requirements

Compute Resources:

Hardware requirements for Bike sharing data analysis can vary depending on the scale and complexity of the analysis you plan to perform. Here's a general outline of hardware requirements for a moderate-sized mall data analysis system:

Server/Compute Resources:

Processor (CPU): Multi-core processors (e.g., Intel Xeon, AMD Ryzen) for parallel processing.

Memory (RAM): Minimum 16GB RAM for basic analysis, but for more extensive data sets and complex algorithms, consider 32GB or more.

Storage: SSDs (Solid State Drives) for fast data access, with sufficient capacity based on the size of the dataset. Consider RAID configurations for data redundancy and improved performance.

Graphics Processing Unit (GPU):

If your data analysis involves machine learning or deep learning algorithms, consider using GPUs (NVIDIA, AMD) to accelerate computations. This is particularly relevant for image recognition, pattern analysis, or any task that benefits from parallel processing.

3.2 Software Requirements

Software requirements for Bike Sharing data analysis encompass a variety of tools and applications that facilitate data processing, analysis, visualization, and management. Here's a list of essential software components:

Data Processing and Analysis:

Programming Languages: Depending on your team's expertise and the nature of the analysis, use languages like Python, R, or Julia for data processing and analysis.

Jupyter Notebooks or RStudio: Interactive environments for developing and sharing code, which can enhance collaboration and documentation.

Data Visualization:

Business Intelligence Tools: Tableau, Power BI, or Qlik for creating interactive and insightful visualizations.

Data Visualization Libraries: Matplotlib, Seaborn, Plotly for Python; ggplot2 for R.

Machine Learning and Statistical Analysis:

Machine Learning Libraries: Scikit-learn, TensorFlow, PyTorch for implementing machine learning algorithms.

Statistical Analysis Software: R for statistical analysis; JASP, SPSS for more advanced statistical modeling.

CHAPTER-4

Product Features

4.1 Screen Shots

In [2]:
```
# mounting drive

from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [84]:
```
# Importing the libraries

import numpy as np
import pandas as pd
from numpy import math
import seaborn as sns
%matplotlib inline
import warnings

from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import linear_model
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error

import matplotlib.pyplot as plt
```

In [98]:
```
# reading data
df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/projects/CP-2 Supervised ML - Regression/dat f
```

Overview of data

In [ ]:
```
# head of data
df.head()
```

Out[ ]:

	Date	Rented Bike Count	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Dew point temperature(°C)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfa (cm
0	01/12/2017	254	0	-5.2	37	2.2	2000	-17.6	0.0	0.0	0.
1	01/12/2017	204	1	-5.5	38	0.8	2000	-17.6	0.0	0.0	0.
2	01/12/2017	173	2	-6.0	39	1.0	2000	-17.7	0.0	0.0	0.
3	01/12/2017	107	3	-6.2	40	0.9	2000	-17.6	0.0	0.0	0.
4	01/12/2017	78	4	-6.0	36	2.3	2000	-18.6	0.0	0.0	0.

In [ ]:
```
# tail of data

df.tail()
```

Out[ ]:

	Date	Rented Bike Count	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Dew point temperature(°C)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snc
8755	30/11/2018	1003	19	4.2	34	2.6	1894	-10.3	0.0	0.0	
8756	30/11/2018	764	20	3.4	37	2.3	2000	-9.9	0.0	0.0	
8757	30/11/2018	694	21	2.6	39	0.3	1968	-9.9	0.0	0.0	
8758	30/11/2018	712	22	2.1	41	1.0	1859	-9.8	0.0	0.0	
8759	30/11/2018	584	23	1.9	43	1.3	1909	-9.3	0.0	0.0	

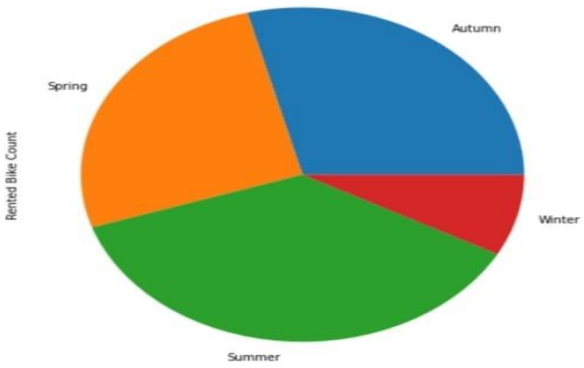
In [ ]:
```
# description of data

df.describe()
```

Out[ ]:

	Rented Bike Count	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Dew point temperature(°C)	Solar Radiation (MJ/m2)	R
count	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	8760.000000	87
mean	704.602055	11.500000	12.882922	58.226256	1.724909	1436.825799	4.073813	0.569111	
std	644.997468	6.922582	11.944825	20.362413	1.036300	608.298712	13.060369	0.868746	
min	0.000000	0.000000	-17.800000	0.000000	0.000000	27.000000	-30.600000	0.000000	
25%	191.000000	5.750000	3.500000	42.000000	0.900000	940.000000	-4.700000	0.000000	
50%	504.500000	11.500000	13.700000	57.000000	1.500000	1698.000000	5.100000	0.010000	
75%	1065.250000	17.250000	22.500000	74.000000	2.300000	2000.000000	14.800000	0.930000	
max	3556.000000	23.000000	39.400000	98.000000	7.400000	2000.000000	27.200000	3.520000	

In [79]:
```
# creating a pie chart of bike count in differant seasons
df_s['Rented Bike Count'].plot(kind='pie', subplots=True, figsize=(8, 8))
```

Out[79]:
```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x7f984ee832d0>],
      dtype=object)
```



conclusions from above pie chart:

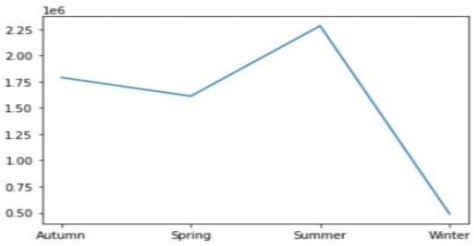
1. most bikes have been rented in the summer season.
2. least bike rent count is in winter season.
3. autumn and spring seasons have almost equal amounts of bike rent count.

In [80]:
```
# creating a dataframe which contains rented bike counts in each seasaon

df_seasons = df.groupby('Seasons').sum()['Rented Bike Count']
```

In [81]:
```
df_seasons.head()
```

Out[81]:
```
Seasons
Autumn    1790002
Spring    1611909
Summer    2283234
Winter     487169
Name: Rented Bike Count, dtype: int64
```

In [ ]:
```
# Line plot showing the differance in rent rate in differant seasons

plt.plot(df_seasons)
plt.show()
```



In [ ]:
```
# creating a series which shows total number of bikes rented in each year

df_year = df.groupby('year').sum()['Rented Bike Count']
```

In [ ]:
```
df_year
```

Out[ ]:
```
year
2017     185330
2018    5986984
Name: Rented Bike Count, dtype: int64
```

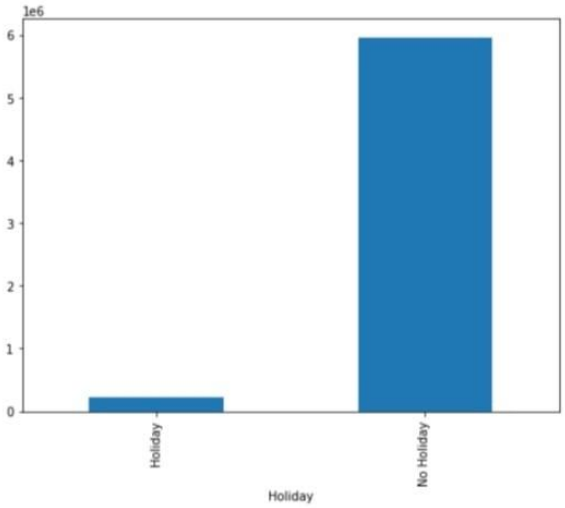
In [ ]:
```
# creating a series which shows total number of bikes rented on the type of day

df_hol = df.groupby('Holiday').sum()['Rented Bike Count']
```

In [ ]:
```
df_hol
```

Out[ ]:
```
Holiday
Holiday        215895
No Holiday    5956419
Name: Rented Bike Count, dtype: int64
```

In [ ]:
```
fig, ax = plt.subplots(figsize=(8,6))
df_hol.plot(kind='bar', ax=ax)
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa448f40450>
```



Above plot shows that most of the bikes have been rented on working days.

In [105…
```
# finding the inter-quartile range

Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1
print(IQR)
```

```
Rented Bike Count             874.25
Hour                           11.50
Temperature(°C)                19.00
Humidity(%)                    32.00
Wind speed (m/s)                1.40
Visibility (10m)             1060.00
Dew point temperature(°C)      19.50
Solar Radiation (MJ/m2)         0.93
Rainfall(mm)                    0.00
Snowfall (cm)                   0.00
year                            0.00
month                           6.00
Winter                          0.00
Spring                          1.00
Summer                          1.00
Autumn                          0.00
dtype: float64
```

In [105…

```
# finding the inter-quartile range

Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1
print(IQR)
```

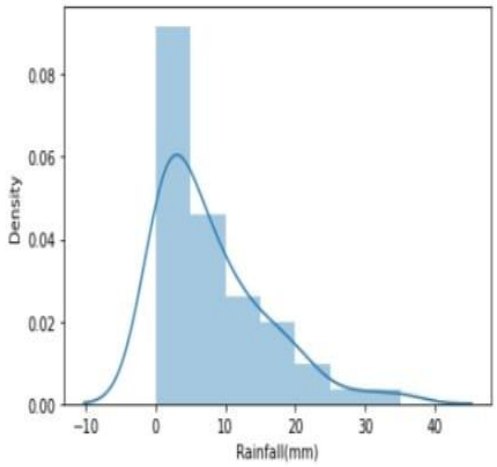
```
Rented Bike Count             874.25
Hour                           11.50
Temperature(°C)                19.00
Humidity(%)                    32.00
Wind speed (m/s)                1.40
Visibility (10m)             1060.00
Dew point temperature(°C)      19.50
Solar Radiation (MJ/m2)         0.93
Rainfall(mm)                    0.00
Snowfall (cm)                   0.00
year                            0.00
month                           6.00
Winter                          0.00
Spring                          1.00
Summer                          1.00
Autumn                          0.00
dtype: float64
```

In [ ]:

```
# plot showing distribution of bike rentals according to rainfall intensity

sns.distplot(df_rain['Rainfall(mm)'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a depr
ecated function and will be removed in a future version. Please adapt your code to use either `displot`
(a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc0c1617b10>

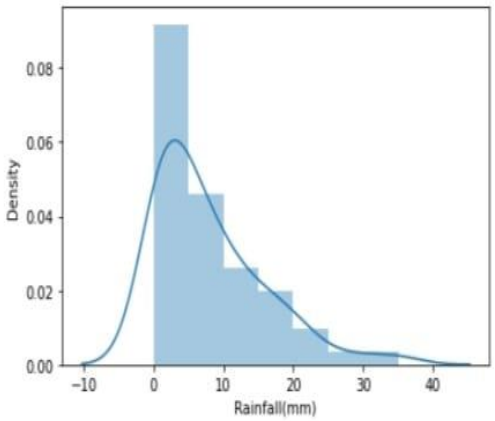


Above plot shows that people tend to rent bikes when there is no or less rainfall.

In [ ]:
```
# plot showing distribution of bike rentals according to rainfall intensity

sns.distplot(df_rain['Rainfall(mm)'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a depr
ecated function and will be removed in a future version. Please adapt your code to use either `displot`
(a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc0c1617b10>

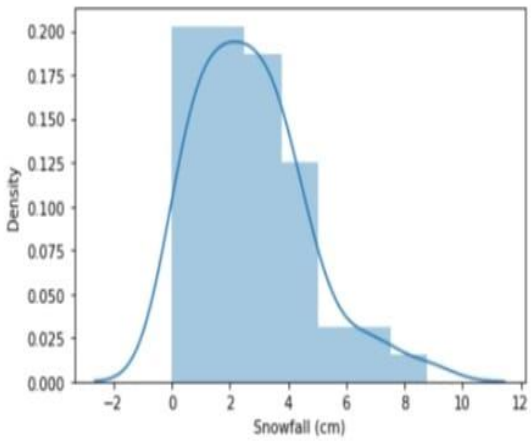


Above plot shows that people tend to rent bikes when there is no or less rainfall.

In [ ]:
```
# plot showing distribution of bike rentals according to snowfall intensity

sns.distplot(df_snow['Snowfall (cm)'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a depr
ecated function and will be removed in a future version. Please adapt your code to use either `displot`
(a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc0c14e8750>

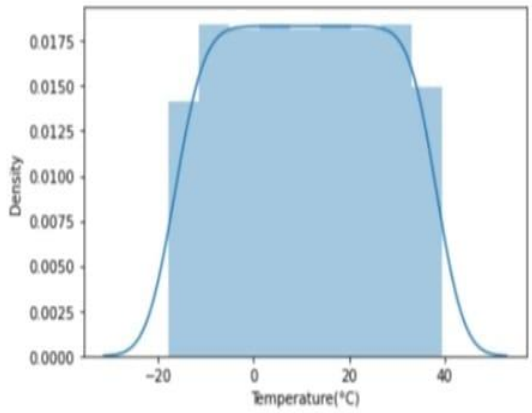


Above plot shows that people tend to rent bikes when there is no or less snowfall.

In [ ]:
```
# plot showing distribution of bike rentals according to temperature intensity

sns.distplot(df_temp['Temperature(°C)'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a depr
ecated function and will be removed in a future version. Please adapt your code to use either `displot`
(a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc0c19b2a90>

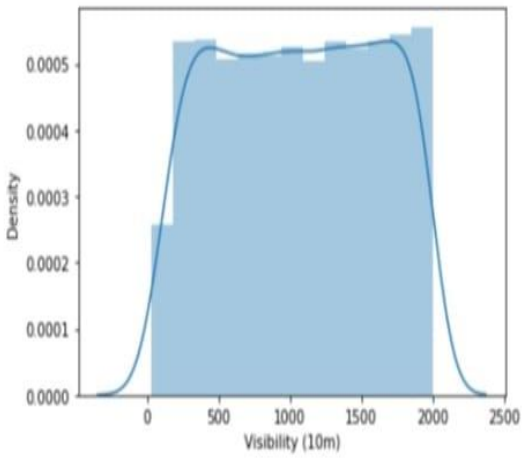


Above plot shows that people tend to rent bikes when the temperature is between -5 to 25 degrees.

In [ ]:
```
sns.distplot(df_visi['Visibility (10m)'])
```

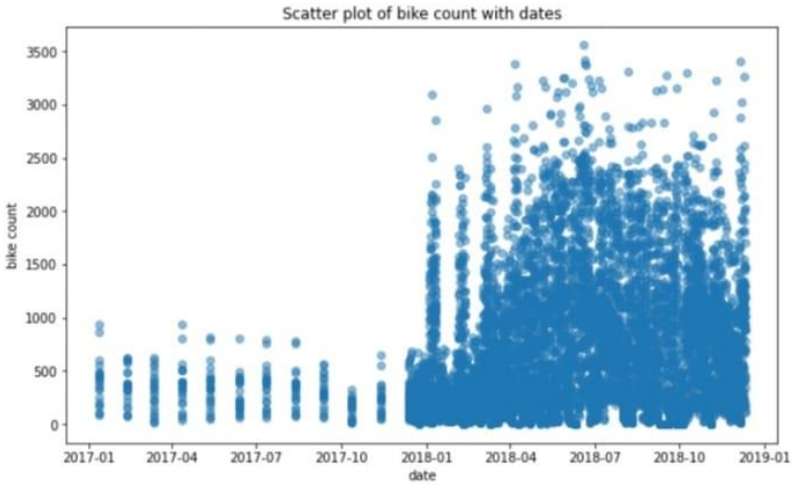
```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a depr
ecated function and will be removed in a future version. Please adapt your code to use either `displot`
(a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7fc0c1400690>



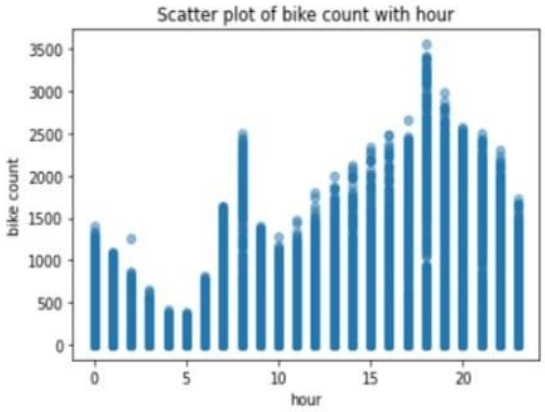
Above plot shows that people tend to rent bikes when the visibility is between 300 to 1700.

In [ ]:
```
# scatter plot of bike count on differant dates

plt.figure(figsize=(10,6))
plt.scatter(df['Date'], df['Rented Bike Count'], alpha=0.5)
plt.title('Scatter plot of bike count with dates')
plt.xlabel('date')
plt.ylabel('bike count')
plt.show()
```



Its evident from above plot that rentals increased in year 2018

In [ ]:
```
# scatter plot of bike count at hour of a particular day

plt.scatter(df['Hour'], df['Rented Bike Count'], alpha=0.5)
plt.title('Scatter plot of bike count with hour')
plt.xlabel('hour')
plt.ylabel('bike count')
plt.show()
```



From above its clear that the rentals were more in the morning and evening.This is because people not having personal vehicle, commuting to offices and schools tend to rent bikes.

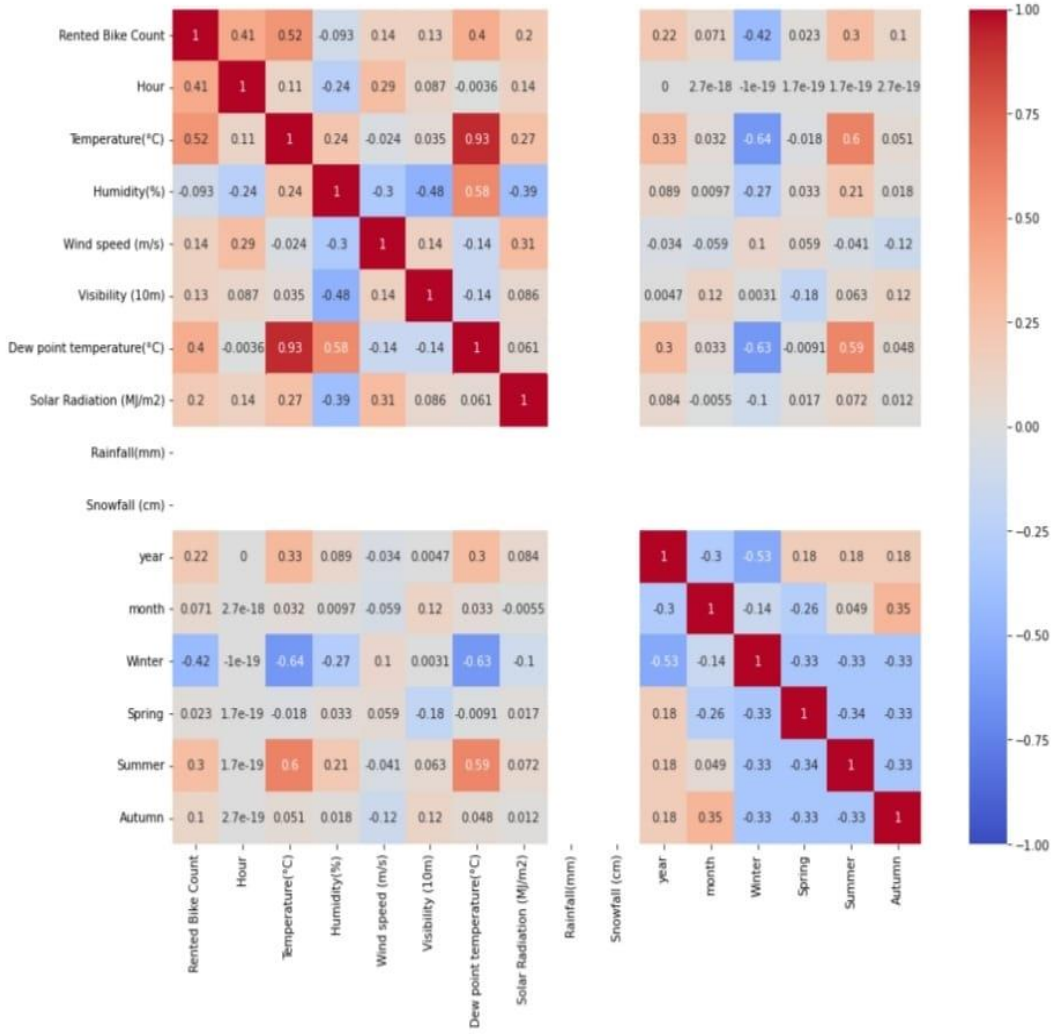
In [110…

```
# filling null values with mean values

df['Temperature(°C)'] = df['Temperature(°C)'].fillna(df['Temperature(°C)'].mean())
df['Humidity(%)'] = df['Humidity(%)'].fillna(df['Humidity(%)'].mean())
df['Wind speed (m/s)'] = df['Wind speed (m/s)'].fillna(df['Wind speed (m/s)'].mean())
df['Visibility (10m)'] = df['Visibility (10m)'].fillna(df['Visibility (10m)'].mean())
df['Dew point temperature(°C)'] = df['Dew point temperature(°C)'].fillna(df['Dew point temperature(°C)'
df['Solar Radiation (MJ/m2)'] = df['Solar Radiation (MJ/m2)'].fillna(df['Solar Radiation (MJ/m2)'].mean
df['Rainfall(mm)'] = df['Rainfall(mm)'].fillna(df['Rainfall(mm)'].mean())
df['Snowfall (cm)'] = df['Snowfall (cm)'].fillna(df['Snowfall (cm)'].mean())
```

In [111…

```
# extracting correlation heatmap

plt.figure(figsize=(15,12))
sns.heatmap(df.corr('pearson'),vmin=-1, vmax=1,cmap='coolwarm',annot=True, square=True)
```

Out[111… <matplotlib.axes._subplots.AxesSubplot at 0x7f9850e8a650>



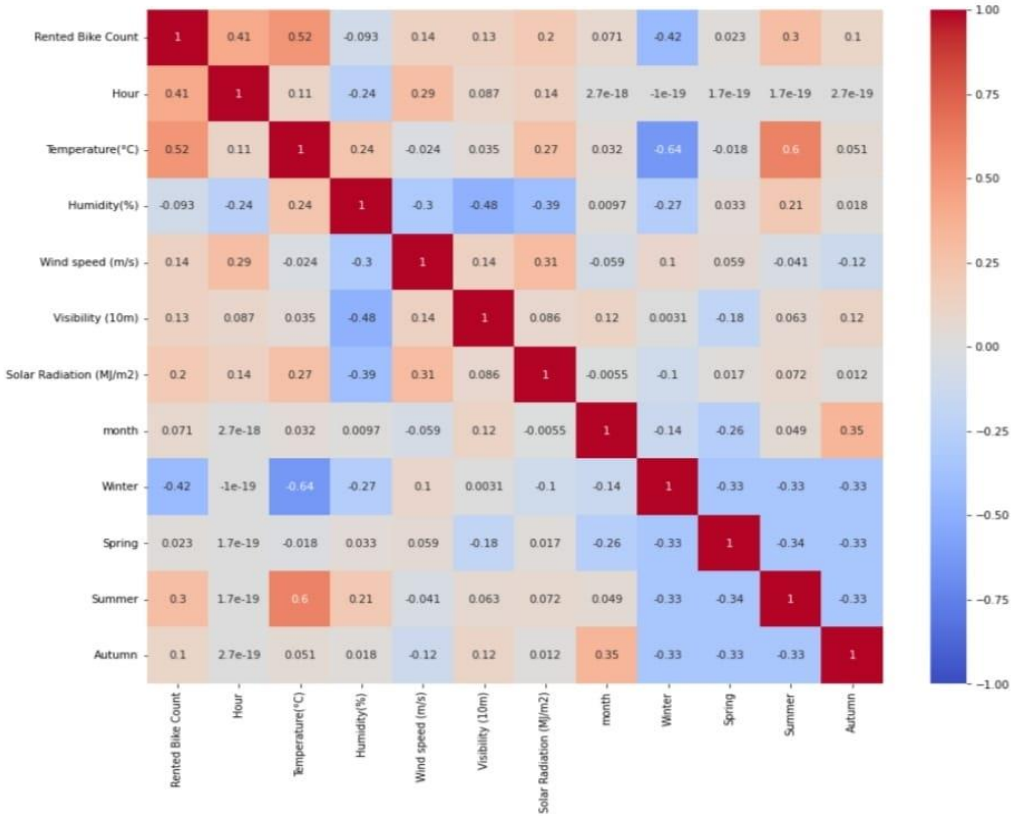
In [112…

```
# dropping columns with more (or less) correlation

df.drop(columns=['Dew point temperature(°C)','Date','Rainfall(mm)','Snowfall (cm)','year'],axis=1,inpla
```

In [113…

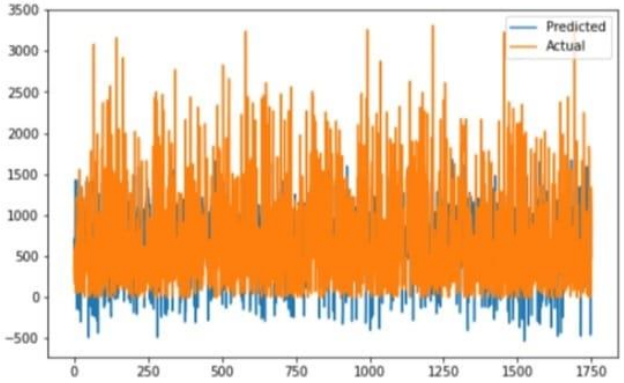
```
# extracting correlation heatmap

plt.figure(figsize=(15,12))
sns.heatmap(df.corr('pearson'),vmin=-1, vmax=1,cmap='coolwarm',annot=True, square=True)
```

Out[113… <matplotlib.axes._subplots.AxesSubplot at 0x7f9854f253d0>



In [35]:

```
# plotting results from above model

plt.figure(figsize=(8,5))
plt.plot((Y_pred_test))
plt.plot(np.array((Y_test)))
plt.legend(["Predicted","Actual"])
plt.show()
```



CHAPTER 5

Code

```
# Importing the libraries
import numpy as np
import pandas as pd
from numpy import math
import seaborn as sns
%matplotlib inline
import warnings
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import linear_model
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
# reading data
df = pd.read_csv("C:\\Users\\lenovo\\Downloads\\bike sharing.csv", encoding='latin1')
# head of data
df.head()
# tail of data
df.tail()
# description of data
df.describe()
# columns in our dataset
df.columns
```

```
df.info()
df.isnull().sum()
# finding the type of data of 'Date' column
type(df['Date'][0])
# converting string format of 'Date' column into date-time format
df['Date'] = pd.to_datetime(df['Date'])
# all the seasons present in data
df['Seasons'].unique()
# creating a column containing the year from a particular date
year = []
for i in range(len(df['Date'])):
  year.append(df['Date'][i].year)
df['year'] = year
# creating a column containing the month number from a particular date
months = []
for i in range(len(df['Date'])):
  months.append(df['Date'][i].month)
df['month'] = months
# creating a dataframe containing the sum of all column values pertaining to differant seasons
df_s = df.groupby('Seasons').sum()
df_s
# creating a pie chart of bike count in differant seasons
df_s['Rented Bike Count'].plot(kind='pie', subplots=True, figsize=(8, 8))
# creating a dataframe which contains rented bike counts in each seasaon
df_seasons = df.groupby('Seasons').sum()['Rented Bike Count']
df_seasons.head()
```

```
# Line plot showing the differance in rent rate in differant seasons

plt.plot(df_seasons)

plt.show()

# creating a series which shows total number of bikes rented in each year

df_year = df.groupby('year').sum()['Rented Bike Count']

df_year

fig, ax = plt.subplots(figsize=(8,6))

df_year.plot(kind='bar', ax=ax)

# creating a series which shows total number of bikes rented on the type of day

df_hol = df.groupby('Holiday').sum()['Rented Bike Count']

df_hol

fig, ax = plt.subplots(figsize=(8,6))

df_hol.plot(kind='bar', ax=ax)

# bikes rented on type of day in each year

plt.figure(figsize=(8,7))

sns.countplot(x='Holiday', hue = 'year', data= df)

plt.title('Density of rented bikes on holiday vs no holiday in a particular year')

plt.show()

# creating a dataframe containing the count of bikes rented in differant intensities of rainfalldf_rain =
pd.DataFrame(df.groupby('Rainfall(mm)')['Rented Bike Count'].sum())

# resetting index of the dataframe

df_rain.reset_index(inplace=True)

df_rain.head()

# plot showing distribution of bike rentals according to rainfall intensity

sns.distplot(df_rain['Rainfall(mm)'])

/usr/local/lib/python3/dist-packages/seaborn/dis
```

```
# creating a dataframe containing the count of bikes rented in differant intensities of snowfall
df_snow = pd.DataFrame(df.groupby('Snowfall (cm)')['Rented Bike Count'].sum())
df_snow.reset_index(inplace=True)
df_snow.head()
# plot showing distribution of bike rentals according to snowfall intensity
sns.distplot(df_snow['Snowfall (cm)'])
# creating a dataframe containing the count of bikes rented in differant intensities of rainfall
df_temp = pd.DataFrame(df.groupby('Temperature(°C)')['Rented Bike Count'].sum())
df_temp.reset_index(inplace=True)
df_temp.head()
# plot showing distribution of bike rentals according to temperature intensity
sns.distplot(df_temp['Temperature(°C)'])
# creating a dataframe containing the count of bikes rented in differant visibility ranges
df_visi = pd.DataFrame(df.groupby('Visibility (10m)')['Rented Bike Count'].sum())
df_visi.reset_index(inplace=True)
df_visi.head()
sns.distplot(df_visi['Visibility (10m)'])
# encoding the season names
df['Winter'] = np.where(df['Seasons']=='Winter', 1, 0)
df['Spring'] = np.where(df['Seasons']=='Spring', 1, 0)
df['Summer'] = np.where(df['Seasons']=='Summer', 1, 0)
df['Autumn'] = np.where(df['Seasons']=='Autumn', 1, 0)
df.drop(columns=['Seasons'],axis=1,inplace=True)
# encoding 'Holiday' column with 0 and 1
```

```
for i in range(len(df['Holiday'])):
  if df['Holiday'][i] == 'No Holiday':
    df['Holiday'][i] = 0
  else:
    df['Holiday'][i] = 1
# encoding 'Functioning Day' column with 0 and 1
for i in range(len(df['Functioning Day'])):
  if df['Functioning Day'][i] == 'Yes':
    df['Functioning Day'][i] = 1
  else:
    df['Functioning Day'][i] = 0
df.head()
# scatter plot of bike count on differant dates
plt.figure(figsize=(10,6))
plt.scatter(df['Date'], df['Rented Bike Count'], alpha=0.5)
plt.title('Scatter plot of bike count with dates')
plt.xlabel('date')
plt.ylabel('bike count')
plt.show()
# scatter plot of bike count at hour of a particular day
plt.scatter(df['Hour'], df['Rented Bike Count'], alpha=0.5)
plt.title('Scatter plot of bike count with hour')
plt.xlabel('hour')
```

```
plt.ylabel('bike count')
plt.show()
# finding the inter-quartile range
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1
print(IQR)
# listing features to remove outliers
features = list(df.columns)
features = features[2:]
list_0 = ['Hour','Winter','Spring','Summer','Autumn','Holiday','Functioning Day','month','year']
new_features = [x for x in features if x not in list_0]
new_features
# removing outliers
df[new_features] = df[new_features][~((df[new_features] < (Q1 - 1.5 * IQR)) |(df[new_features] > (Q3 + 1.5 * IQR))).any(axis=1)]
df.info()
# extracting correlation heatmap
plt.figure(figsize=(15,12))
sns.heatmap(df.corr('pearson'),vmin=-1, vmax=1,cmap='coolwarm',annot=True, square=True)
# extracting correlation heatmap
plt.figure(figsize=(15,12))
sns.heatmap(df.corr('pearson'),vmin=-1, vmax=1,cmap='coolwarm',annot=True, square=True)
# function to calculate Multicollinearity
from statsmodels.stats.outliers_influence import variance_inflation_factor
def calc_vif(X):
  # Calculating VIF
```

```
    vif = pd.DataFrame()
    vif["variables"] = X.columns
    vif["VIF"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
  return(vif)
# multicollinearity result
calc_vif(df[[i for i in df.describe().columns if i not in ['Rented Bike Count','Date']]])
# dropping "summer" column as it adds to multicollinearity
df.drop(columns=['Summer'],axis=1,inplace=True)
calc_vif(df[[i for i in df.describe().columns if i not in ['Rented Bike Count','Date']]])
numeric_features = df.columns
for col in numeric_features[1:]:
    fig = plt.figure(figsize=(9, 6))
    ax = fig.gca()
    feature = df[col]
    label = df['Rented Bike Count']
    correlation = feature.corr(label)
    plt.scatter(x=feature, y=label)
    plt.xlabel(col)
    plt.ylabel('Rented Bike Count')
    ax.set_title('Rented Bike Count vs ' + col + '- correlation: ' + str(correlation))
    z = np.polyfit(df[col], df['Rented Bike Count'], 1)
    y_hat = np.poly1d(z)(df[col])
 plt.plot(df[col], y_hat, "r--", lw=1)
plt.show()
regressor = LinearRegression()
regressor.fit(X_train, Y_train)
# Predicting the Train set results
```

```
Y_pred_train = regressor.predict(X_train)

# Predicting the Test set results

Y_pred_test = regressor.predict(X_test)

# r2 score of train set

r2_linear_train = r2_score(Y_train, Y_pred_train)

r2_linear_train

r2_linear_test = r2_score(Y_test, Y_pred_test)

# plotting results from above model

plt.figure(figsize=(8,5))

plt.plot((Y_pred_test))

plt.plot(np.array((Y_test)))

plt.legend(["Predicted","Actual"])

plt.show()

# training model

from sklearn.tree import DecisionTreeRegressor

from sklearn.model_selection import GridSearchCV

decisionTree = DecisionTreeRegressor()

param = {'max_depth' : [1,4,5,6,7,10,15,20,8]}

gridSearch_decisionTree=GridSearchCV(decisionTree,param,scoring='r2',cv=6)

gridSearch_decisionTree.fit(X_train,Y_train)

best_DecisionTree=gridSearch_decisionTree.best_estimator_

bestDecisionTree_testScore=best_DecisionTree.score(X_test,Y_test)

r2_decision_test = best_DecisionTree.score(X_test,Y_test)

# extracting best parameters

print(f"The best Decision Tree R2 score is {gridSearch_decisionTree.best_score_} with max depth
{gridSearch_decisionTree.best_params_['max_depth']}")print('\n')

print(f"The best R2 test score is : {bestDecisionTree_testScore} with max depth =
{gridSearch_decisionTree.best_params_['max_depth']}")
```

CHAPTER-6

Conclusion

We are finally at the conclusion of our project Coming from the beginning we did EDA on the dataset and also cleaned the data according to our needs. After that we were able to draw relevant conclusions from the given data and then we trained our model on linear regression and other models Out of all models used , with extra-trees regression model we were able to get the r2- score of 0.85.The model which performed poorly was elastic net regularization with r2- score of 0.42.Given the size of data and the amount of irrelevance in the data , the above score is good..The prediction of bike sharing demand has become increasingly important with the rise of bike-sharing programs in urban areas worldwide. By analyzing historical data and various factors influencing bike usage, predictive models can forecast future demand accurately, aiding in resource allocation, system optimization, and better user experiences.

CHAPTER-7

References

Javatpoint

Geeks For Geeks

W3school

CHAPTER-8

Future Scope

The future scope of bike sharing prediction is promising, with several potential avenues for further research and application:

- Integration of Real-Time Data: Incorporating real-time data streams from various sources such as weather forecasts, traffic conditions, and special events can enhance the accuracy of bike sharing predictions. Future models may leverage advanced data integration techniques and streaming analytics to adapt to dynamic changes in demand patterns.
- Multimodal Transportation Integration: Future bike sharing prediction models may consider the integration of multimodal transportation systems, including public transit, ride-sharing services, and walking routes. This holistic approach can enable more comprehensive mobility solutions and facilitate seamless intermodal transitions for commuters.

- Sustainability and Environmental Impact: Future research in bike sharing prediction may focus on quantifying the environmental benefits and sustainability impacts of bike sharing systems. Predictive models can assess the reduction in carbon emissions, traffic congestion, and energy consumption attributable to increased bike usage, informing policy decisions and urban planning efforts.
- Urban Planning and Infrastructure Optimization: Bike sharing prediction can inform urban planners and policymakers about optimal locations for bike stations, route planning, and infrastructure investments. Predictive analytics can help identify high-demand areas, optimize resource allocation, and improve the accessibility and usability of bike sharing systems within urban environments.

CHAPTER-9

Bibliography

Fanaee-T, Hadi, and Gama, João. "Event labeling combining ensemble detectors and background knowledge." Progress in Artificial Intelligence, vol. 2, no. 3-4, 2013, pp. 113-127.

Guo, Jingwei, et al. "A survey of data-driven prediction methods for bike-sharing systems." ACM Computing Surveys (CSUR), vol. 51, no. 4, 2019, pp. 1-35.

Ma, Liang, et al. "Short-term bike-sharing demand forecasting using gradient boosting decision trees." Transportation Research Part C: Emerging Technologies, vol. 98, 2019, pp. 289-304.

Zheng, Yu, et al. "Forecasting bike-sharing system behavior using the k-nearest neighbor algorithm." IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 4, 2018, pp. 1192-1201.