

12. Mini Project: Bike Sharing Demand Analysis Regression

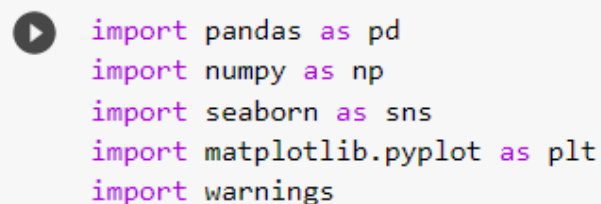
Dataset Information

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, the user is able to easily rent a bike from a particular position and return back to another position. Currently, there are about over 500 bike-sharing programs around the world which are 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. Opposed to other transport services such as bus or subway, the duration of travel, departure, and arrival position is explicitly recorded in these systems. This feature turns the bike-sharing system into a virtual sensor network that can be used for sensing mobility in the city. Hence, it is expected that the most important events in the city could be detected via monitoring these data.

Import modules

These are some modules that we need to import.



```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
```

Loading the dataset

Here I have taken the dataset which consists of today's date, season, year, month, hour, holiday, working day, temp, humidity, etc...

```

✓ [2] df = pd.read_csv('hour.csv')
      df.head()

```

	instant	dteday	season	yr	mnth	hr	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
0	1	2011-01-01	1	0	1	0	0	6	0	1	0.24	0.2879	0.81	0.0	3	13	16
1	2	2011-01-01	1	0	1	1	0	6	0	1	0.22	0.2727	0.80	0.0	8	32	40
2	3	2011-01-01	1	0	1	2	0	6	0	1	0.22	0.2727	0.80	0.0	5	27	32
3	4	2011-01-01	1	0	1	3	0	6	0	1	0.24	0.2879	0.75	0.0	3	10	13
4	5	2011-01-01	1	0	1	4	0	6	0	1	0.24	0.2879	0.75	0.0	0	1	1

Preprocessing the dataset

Here I have preprocessed the data for removing the null values from the dataset. this will use when want to perform cleaning on the dataset.

```

✓ [5] # check for null values
      df.isnull().sum()

```

```

instant      0
dteday       0
season        0
yr            0
mnth          0
hr            0
holiday       0
weekday       0
workingday    0
weathersit     0
temp          0
atemp         0
hum           0
windspeed     0
casual        0
registered    0
cnt           0
dtype: int64

```

Renaming is a part of cleaning and pre-processing the data for better understanding.

```

[6] df = df.rename(columns={'weathersit': 'weather',
                          'yr': 'year',
                          'mnth': 'month',
                          'hr': 'hour',
                          'hum': 'humidity',
                          'cnt': 'count'})
      df.head()

```

	instant	dteday	season	year	month	hour	holiday	weekday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
0	1	2011-01-01	1	0	1	0	0	6	0	1	0.24	0.2879	0.81	0.0	3	13	16
1	2	2011-01-01	1	0	1	1	0	6	0	1	0.22	0.2727	0.80	0.0	8	32	40
2	3	2011-01-01	1	0	1	2	0	6	0	1	0.22	0.2727	0.80	0.0	5	27	32
3	4	2011-01-01	1	0	1	3	0	6	0	1	0.24	0.2879	0.75	0.0	3	10	13
4	5	2011-01-01	1	0	1	4	0	6	0	1	0.24	0.2879	0.75	0.0	0	1	1

Exploratory Data Analysis

Data analysis is important because it gives us a clear view and more understandable information.

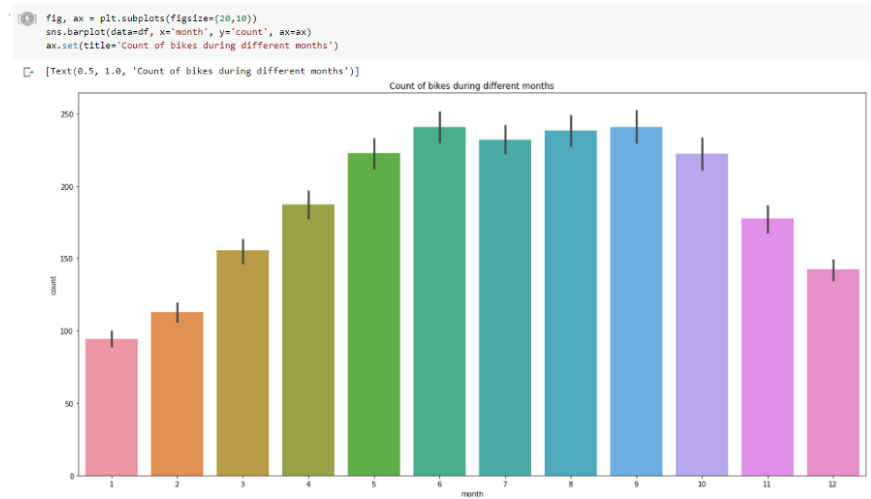
below is a chart that indicates how much bike-sharing is on-demand on weekdays and weekends.



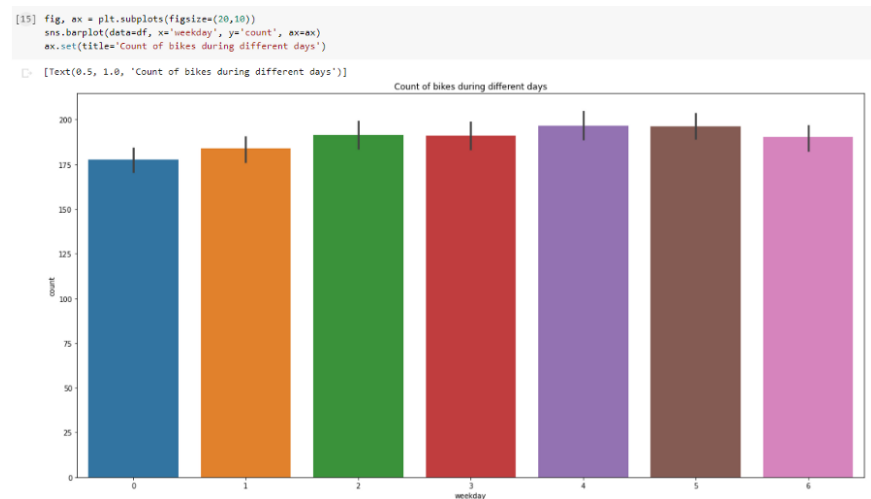
The below figure counts the number of bikes sharing demand during a different season like here I take 1,2,3,4 respecting spring, summer, fall, and winter.



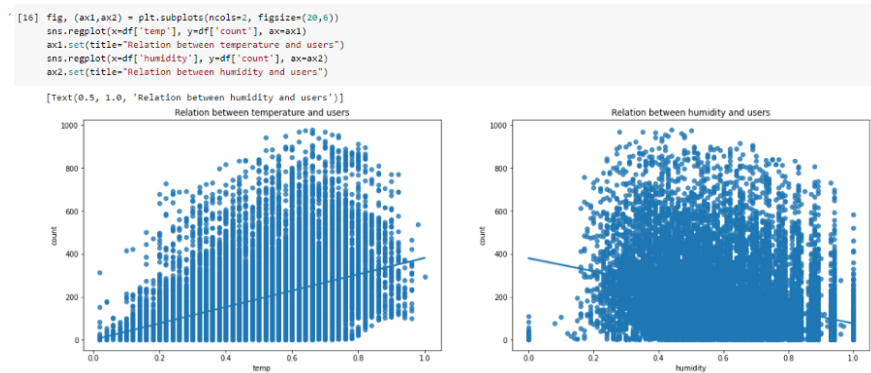
The below figure counts the number of bike-sharing demands during different months. like 1 indicate January, 2- February, 3-march, 4- April like that...



The below figure counts the number of bike-sharing demands during different days of the week. like 0- Sunday, 1- Monday, 2-Tuesday like that...



The below figures shows the relation between humidity and the users



Correlation Matrix

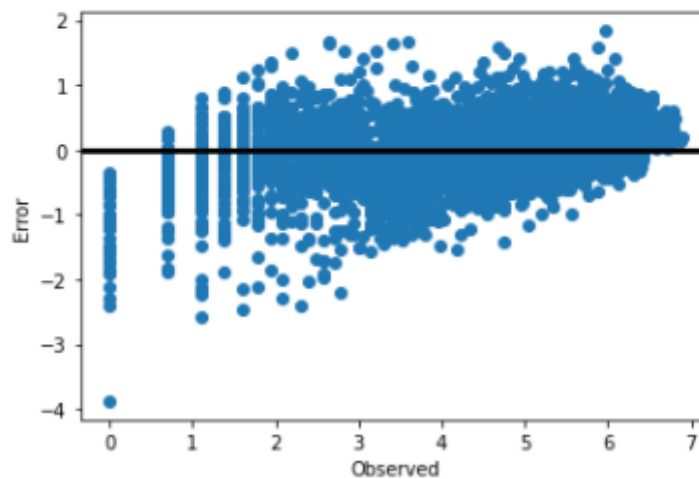
A correlation matrix is a table showing correlation coefficients between variables. Each cell in the table shows the correlation between two variables. A correlation matrix is used **to summarize data, as an input into a more advanced analysis, and as a diagnostic for advanced analyses.**



Input Split and testing the model

splitting the data into testing and training dataset and then make a model and then check that the model that we have created is giving a correct output or not and how much error it has. these are all things we can see while testing the model.

```
# plot the error difference
error = y_test - y_pred
fig, ax = plt.subplots()
ax.scatter(y_test, error)
ax.axhline(lw=3, color='black')
ax.set_xlabel('Observed')
ax.set_ylabel('Error')
plt.show()
```



. . .

That's all for this project. Thank uh.