Bike Sharing

By –

Ganesh Naik Vidhya Pawar Rutika Patil Harib Parol

Vikal Dhawane Rohit Shivramwar Pooja Kharatmol

About the Existing System:

- A bike-sharing system is a service in which bikes are made available for shared use to individuals on a short term basis for a price or free. Many bike share systems allow people to borrow a bike from a "dock" which is usually computer-controlled wherein the user enters the payment information, and the system unlocks it. This bike can then be returned to another dock belonging to the same system.

Need of Proposed System:

A bike-sharing system is a service in which bikes are made available for shared use to individuals on a short term basis for a price or free. Many bike share systems allow people to borrow a bike from a "dock" which is usually computer-controlled wherein the user enters the payment information, and the system unlocks it. This bike can then be returned to another dock belonging to the same system.

The company wants to know -

- 1. Which variables are significant in predicting the demand for shared bikes.
- 2. How well those variables describe the bike demands

Problem Statement -

- A US bike-sharing provider BoomBikes has recently suffered considerable dips in their revenues due to the ongoing Corona pandemic. The company is finding it very difficult to sustain in the current market scenario. So, it has decided to come up with a mindful business plan to be able to accelerate its revenue as soon as the ongoing lockdown comes to an end, and the economy restores to a healthy state.

Scope of the proposed System:

1. Index:- Company can predict the count of bikes which will be Rented depending on the important features.

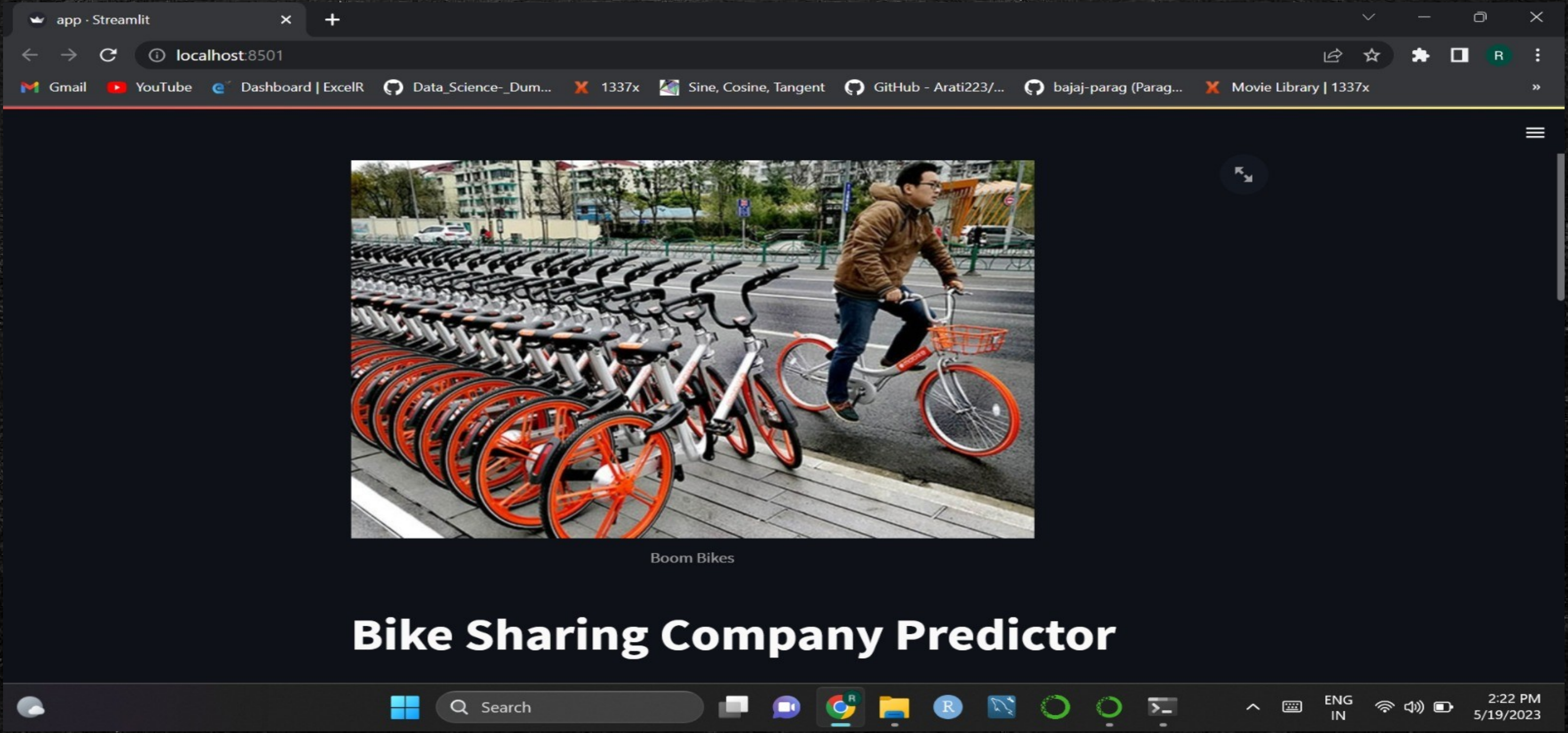
Technologies Used -

- Frontend – HTML, CSS
- Backend – Python, Streamlit
- Tools – Spyder , Jupyter Notebook

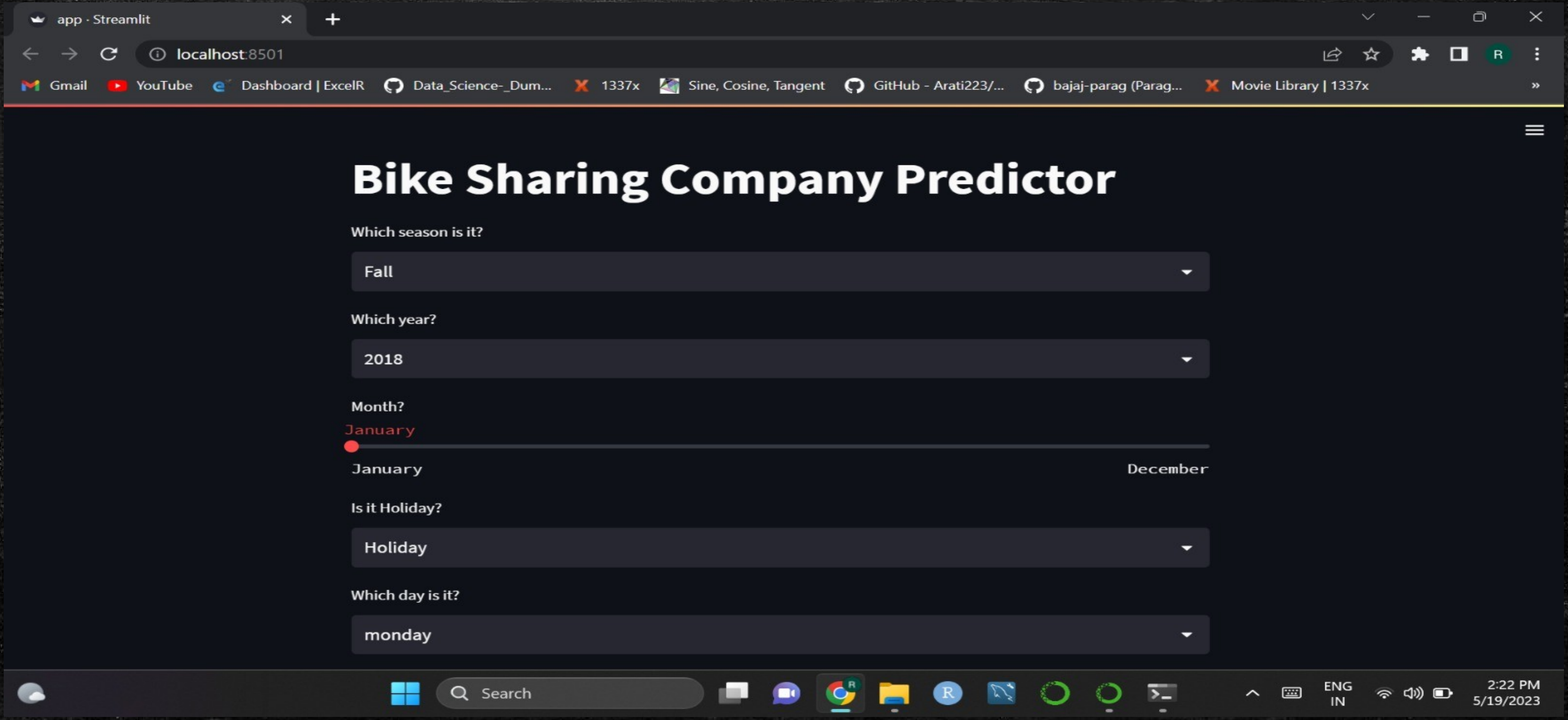
Expected GUI System:

1. Index page

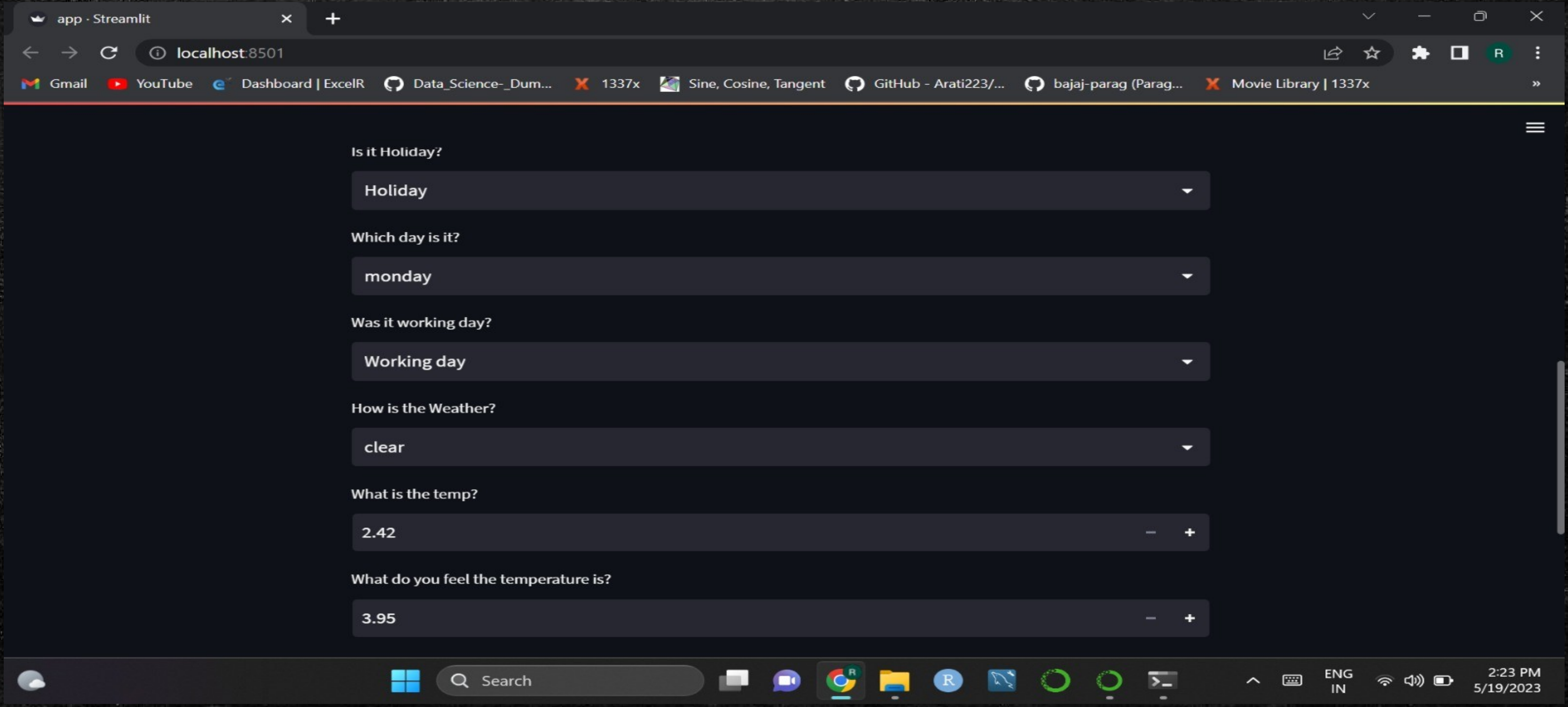
Interface Design :



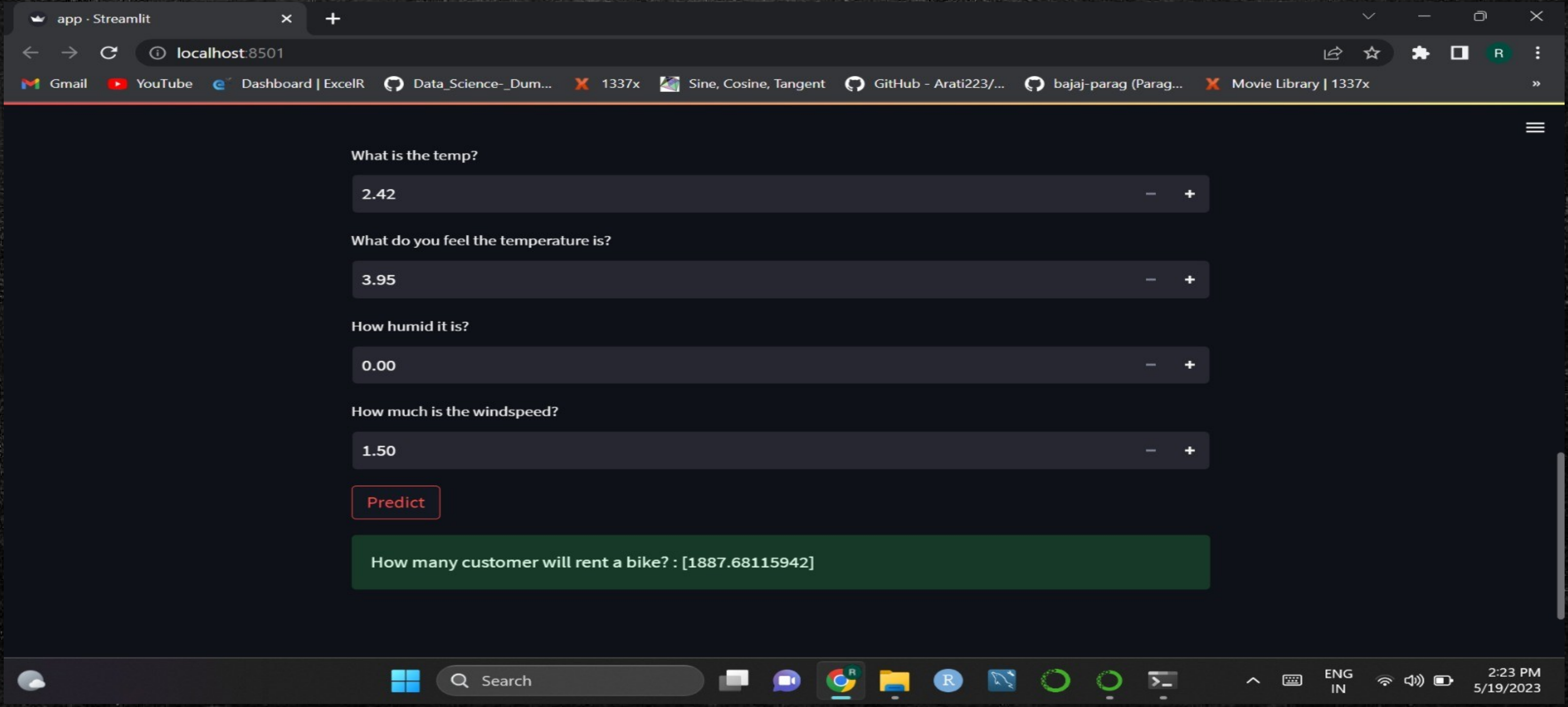
Interface Design :



Interface Design :



Interface Design :



Imported Libraries :

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_style("darkgrid")
import statsmodels.formula.api as smf
from sklearn.feature_selection import RFE
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import MinMaxScaler
from sklearn.cluster import KMeans
from sklearn.preprocessing import PolynomialFeatures
from sklearn.ensemble import BaggingRegressor, RandomForestRegressor
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import SGDRegressor, Lasso, ElasticNet, Ridge
from sklearn.svm import SVR, NuSVR
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
import warnings
from sklearn import svm
from sklearn.metrics import accuracy_score
warnings.filterwarnings("ignore")
from sklearn.model_selection import train_test_split, GridSearchCV, KFold, cross_val_score
from statsmodels.stats.outliers_influence import variance_inflation_factor
pd.set_option('display.max_column', None)
```

Dataset :

```
In [25]: df
```

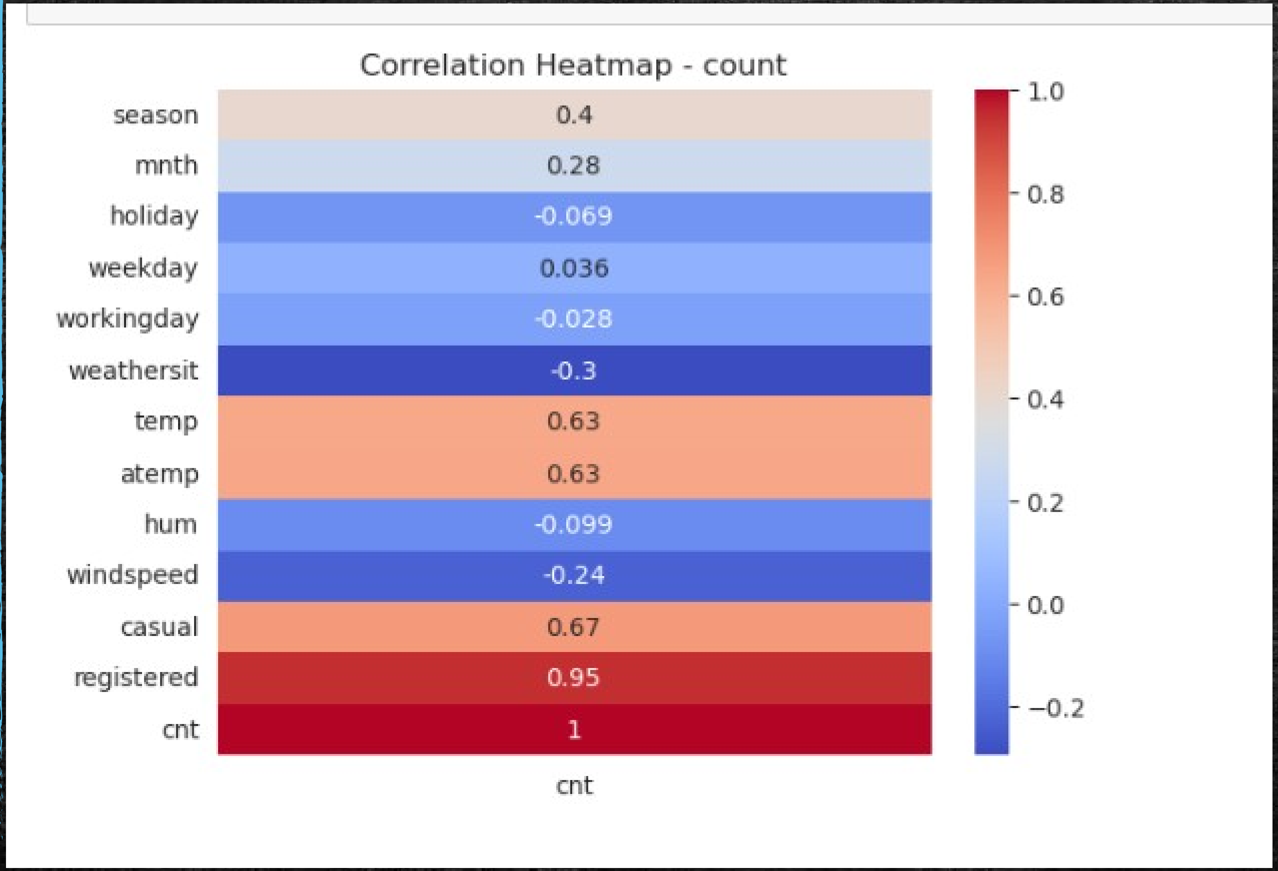
Out[25]:

	season	year	month	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	count
0	Spring	2018	January	not-holiday	monday	working day	misty	14.110847	18.18125	80.5833	10.749882	331	654	985
1	Spring	2018	January	not-holiday	tuesday	working day	misty	14.902598	17.68695	69.6087	16.652113	131	670	801
2	Spring	2018	January	not-holiday	wednesday	working day	clear	8.050924	9.47025	43.7273	16.636703	120	1229	1349
3	Spring	2018	January	not-holiday	thursday	working day	clear	8.200000	10.60610	59.0435	10.739832	108	1454	1562
4	Spring	2018	January	not-holiday	friday	working day	clear	9.305237	11.46350	43.6957	12.522300	82	1518	1600
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
725	Spring	2019	December	not-holiday	friday	working day	misty	10.420847	11.33210	65.2917	23.458911	247	1867	2114
726	Spring	2019	December	not-holiday	saturday	non-working day	misty	10.386653	12.75230	59.0000	10.416557	644	2451	3095
727	Spring	2019	December	not-holiday	sunday	non-working day	misty	10.386653	12.12000	75.2917	8.333661	159	1182	1341
728	Spring	2019	December	not-holiday	monday	working day	clear	10.489153	11.58500	48.3333	23.500518	364	1432	1796
729	Spring	2019	December	not-holiday	tuesday	working day	misty	8.849153	11.17435	57.7500	10.374682	439	2290	2729

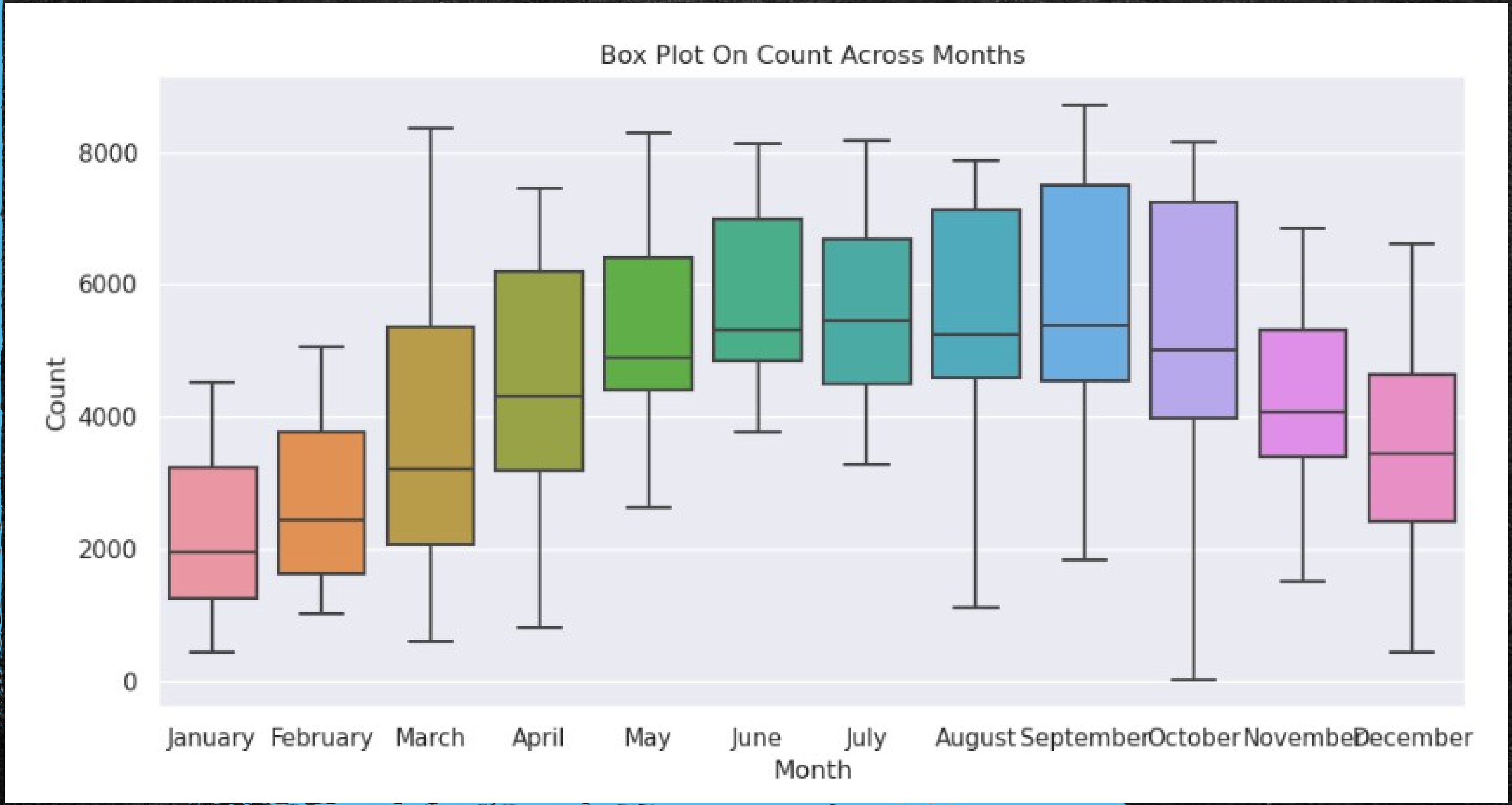
730 rows × 14 columns

Correlation :

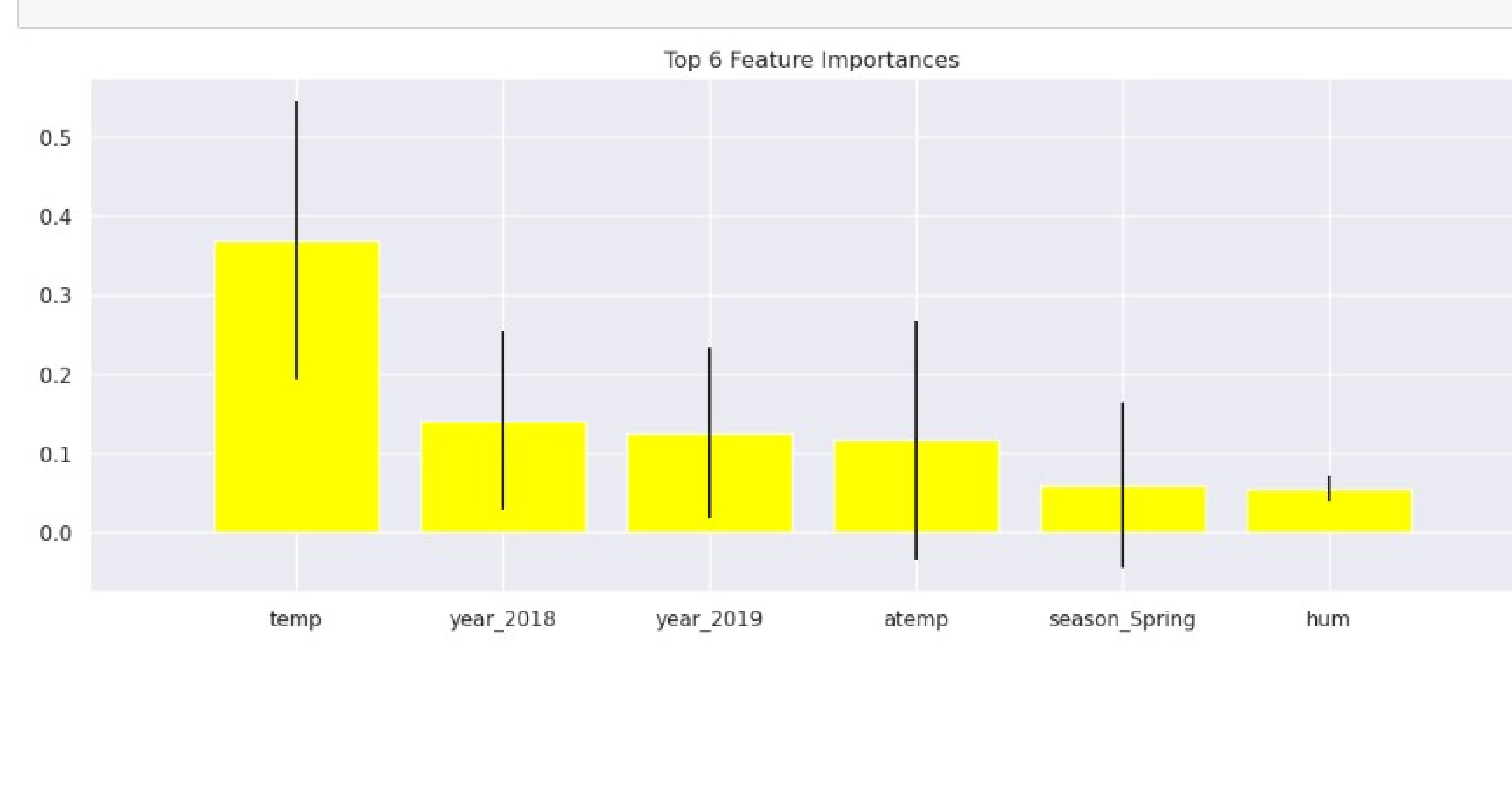
Correlation heatmap with respect to cnt :



Boxplot on count across months :



Top 6 important Features :



Model Prediction :

```
In [130]: def predict(model):
              model= model.fit(xtrain,ytrain)
              ypred= model.predict(xtest)

              print(model.score(xtrain,ytrain))
              print(model.score(xtest,ytest))

In [131]: predict(AdaBoostRegressor(random_state=1))

          0.8424292035616374
          0.8255669027919567

In [132]: predict(GradientBoostingRegressor())

          0.948108952757542
          0.8938354038129837

In [133]: predict(RandomForestRegressor(random_state=1))

          0.9795583555797495
          0.9025567704350393

In [134]: from sklearn.svm import SVC

          predict(SVC())

          0.02910958904109589
          0.0

In [135]: from sklearn.linear_model import LinearRegression

          predict(LinearRegression())

          0.7778788113419504
          0.8537490231887983

In [136]: from sklearn.tree import DecisionTreeRegressor

          predict(DecisionTreeRegressor())

          1.0
          0.8166707085634423
```

Thank You