Bike Share Prediction

Detailed Project Report

Submitted by Sameer Singh

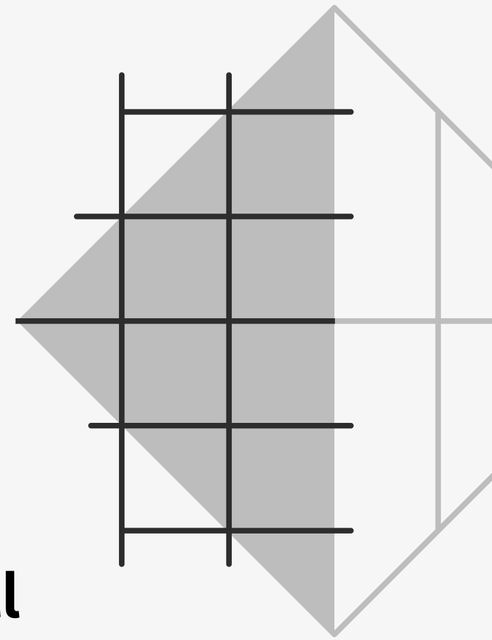
Title	Bike Share Prediction
Technologies:	Machine Learning
Domain:	Transport
Project Difficulty Level	Intermediate
Programmin Language	Python
Tools Used	Vscode, Jupyter Notebook, Github, railway.app

Introduction

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.

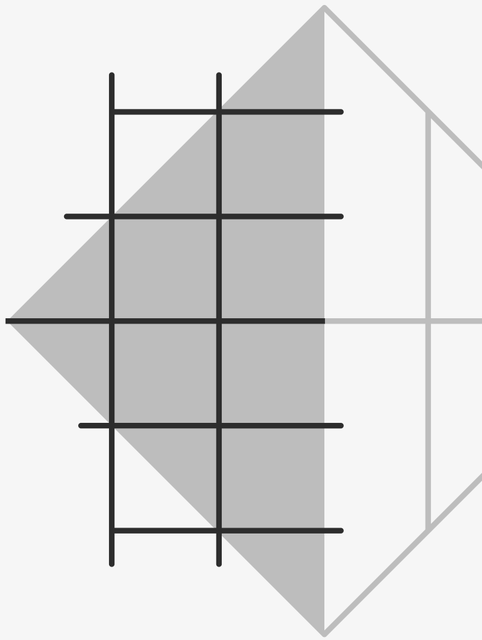
Problem Statement

The most important problem from a business point of view for bike-sharing system like Capital Bikeshare is to predict the bike demand on any particular day. While having excess bikes results in wastage of resources (bike maintenance and 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 this company Capital Bikeshare.



Objective

The goal is to combine the historical Bike usage patterns with the weather data and develop a predictive model for forecasting bike rental demand at a particular hour of a particular day based on different conditions like temperature, humidity level. weather, season, hour of the day, month, and so on.



Requirements

Hardware requirements: A working computer to code with active internet connection.

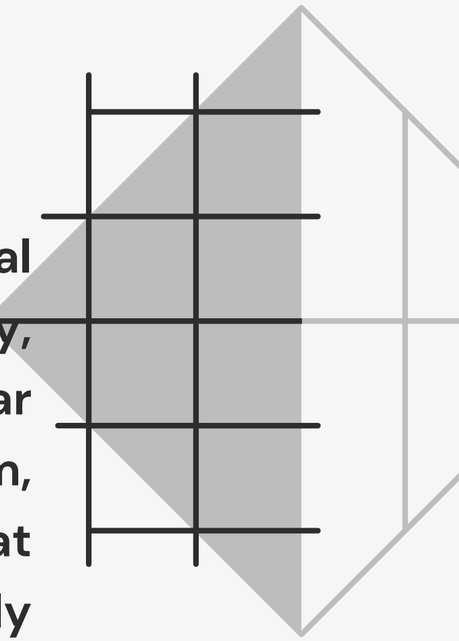
Tools/Software requirements:

- Python has been used for this project.
- Python libraries such as NumPy, pandas, matplotlib, seaborn and scikit-learn (Used for implementation of machine learning algorithms.)
- Jupyter for Exploratory Data Analysis and testing code, Visual studio code is used as an IDE for writing the code.
- HTML, CSS & Java Scripts are used for developing the front end of our web application.
- Flask is used for backend development.
- Github is used as the version control system.
- railway.app is used for deployment.

Data Collection

The dataset was taken from the UCI Machine Learning Repository .
Dataset link hour.csv

Bike-sharing rental process is highly correlated to the environmental and seasonal settings. For instance, weather conditions, day of week, season, hour of the day, etc. can affect the rental behaviors. The core data set is related to the two-year historical log corresponding to years 2011 and 2012 from Capital Bikeshare system, Washington D.C., USA which is publicly available at http://capitalbikeshare.com/system-data. We have taken the data which is hourly based. This data also has data on corresponding weather and seasonal information. Weather
information from http://www.freemeteo.com .



About the Dataset

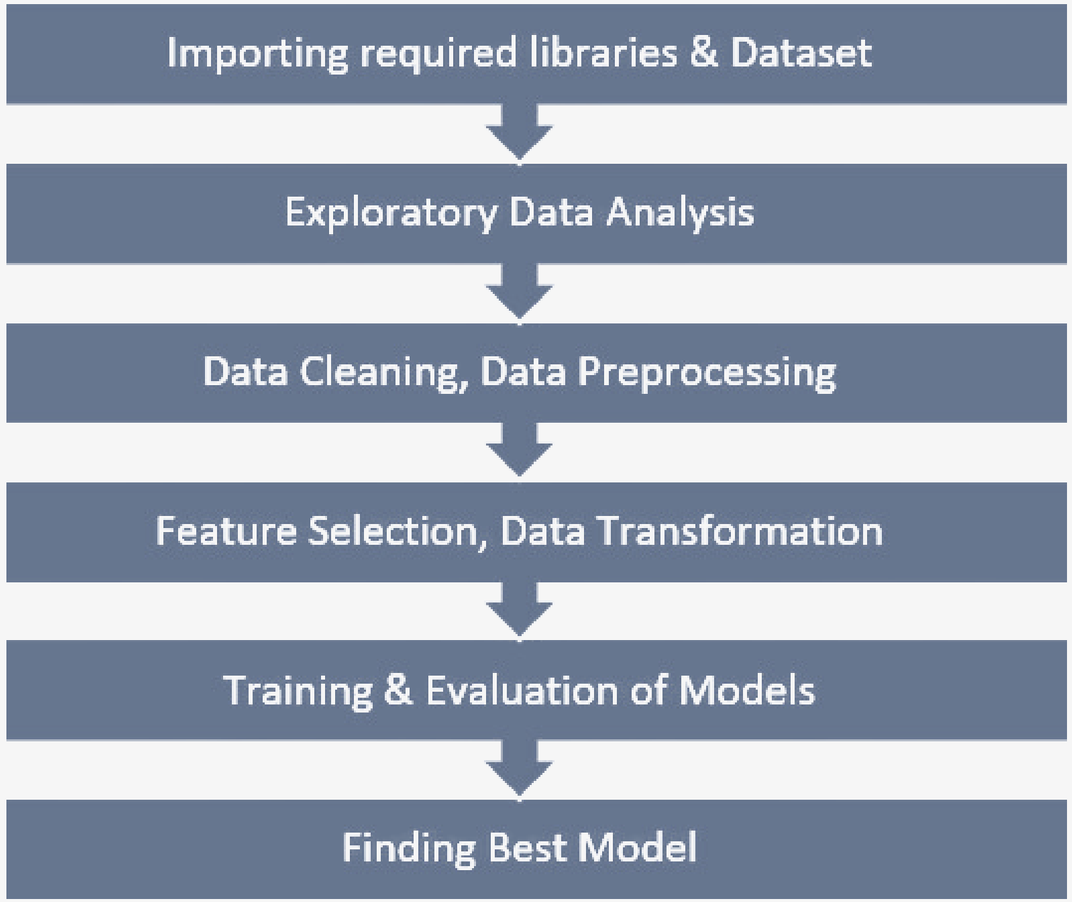
Bike sharing counts aggregated on hourly basis. Records available: 17379 .

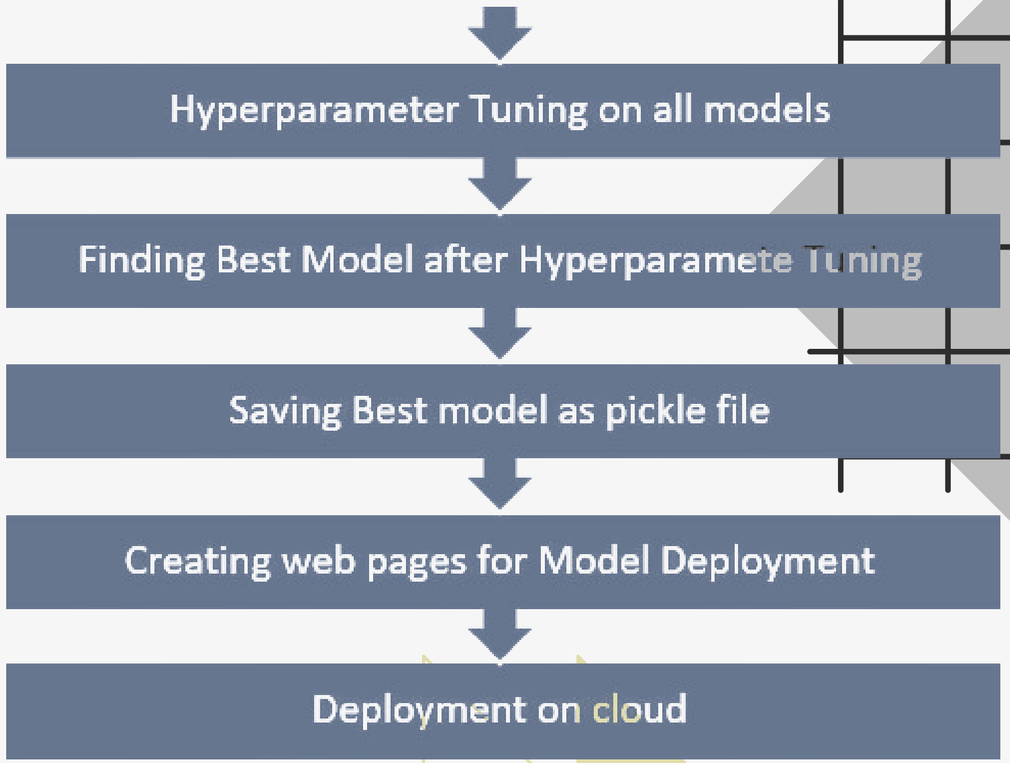
- instant: record index
- dteday : date
- season : season (1:springer, 2:summer, 3:fall, 4:winter)
- yr : year (0: 2011, 1:2012)
- mnth : month (1 to 12)
- hr : hour (0 to 23)
- holiday : weather day is holiday or not (from http://dchr.dc.gov/page/holiday schedule)
- weekday : day of the week
- workingday : if day is neither weekend nor holiday is 1, otherwise is 0.

+ weathersit :

- 1: Clear, Few clouds, Partly cloudy, Partly cloudy
- 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
- 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
- 4: Heavy Rain + Ice Pellets + Thunderstorm + Mist, Snow + Fog
- temp : Normalized temperature in Celsius. The values are divided to 41 (max)
- atemp: Normalized feeling temperature in Celsius. The values are divided to 50 (max)
- hum: Normalized humidity. The values are divided to 100 (max)
- windspeed: Normalized wind speed. The values are divided to 67 (max)
- casual: count of casual users
- registered: count of registered users
- cnt: count of total rental bikes including both casual and registered

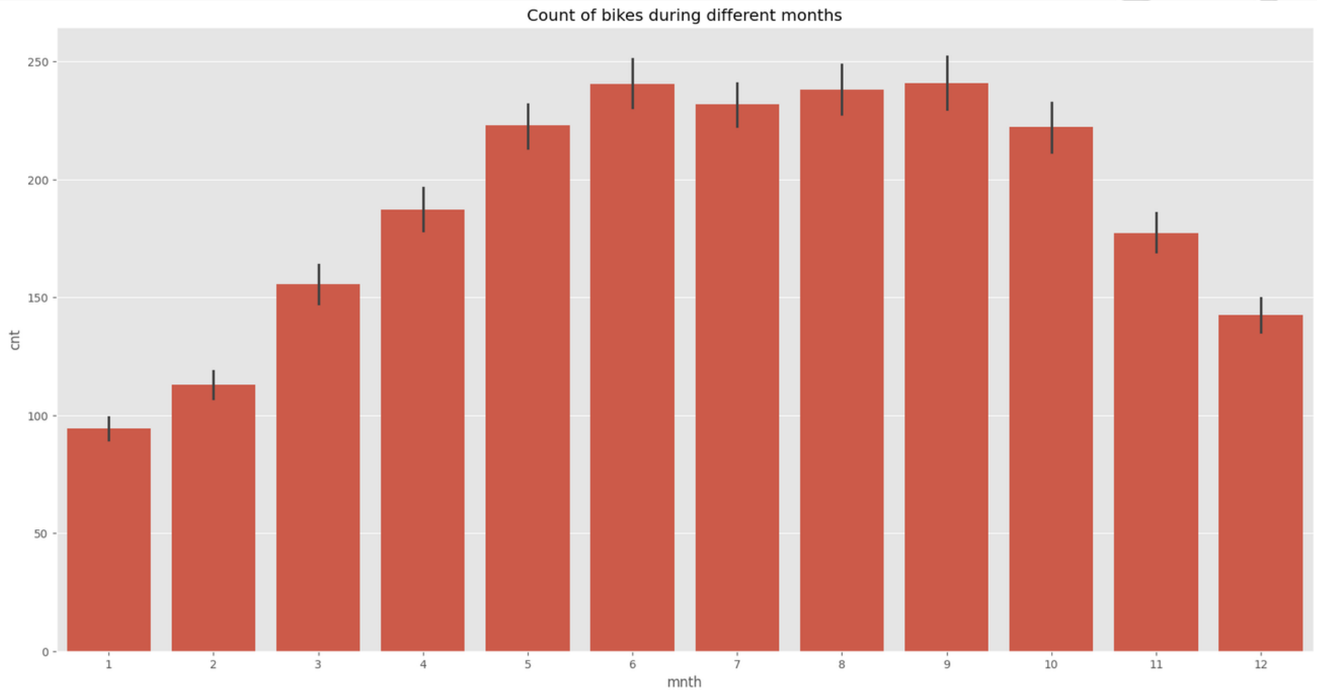
Architecture Design





Exploratory Data Analysis (EDA)

Monthly Distribution



The trend of increasing use of bike starts from january(lowest) till June then stays almost the same till september and then starts dropping. There's a scope to increase the bike usage in the months from january till May and from October to december. The drop of bike usage from october till December might be explained by the winter season and less bike usage from January to April might be explained by higher windspeed.

Weather

▶**Higher bike rental when weather is more clear and sunny**

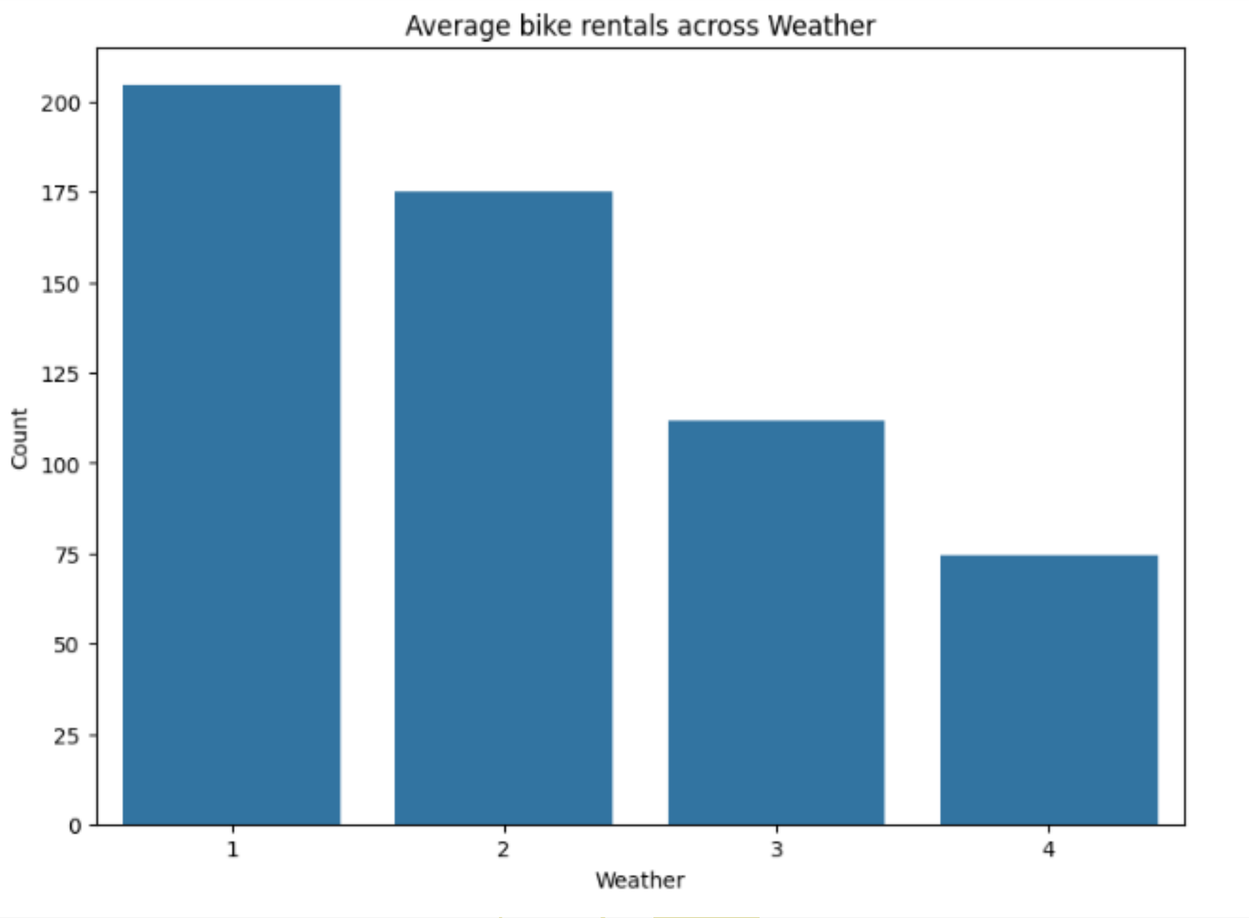
For the given bar plot for weather axis
+ x axis values mean the following:

- 1: Clear, Few clouds, Partly cloudy

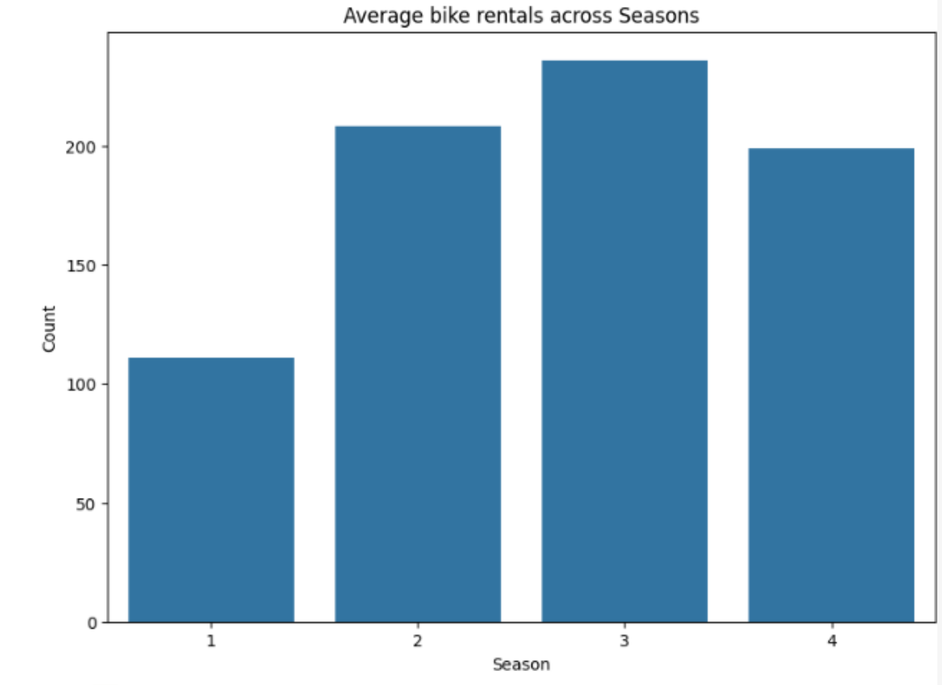
- 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist

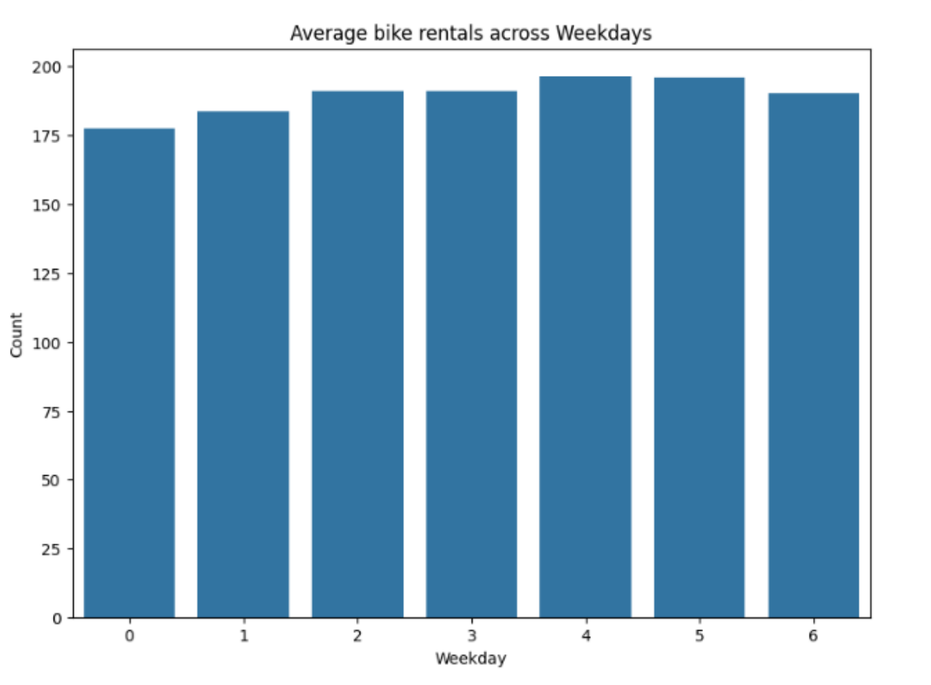
- 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds

- 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog



Season & Weekdays



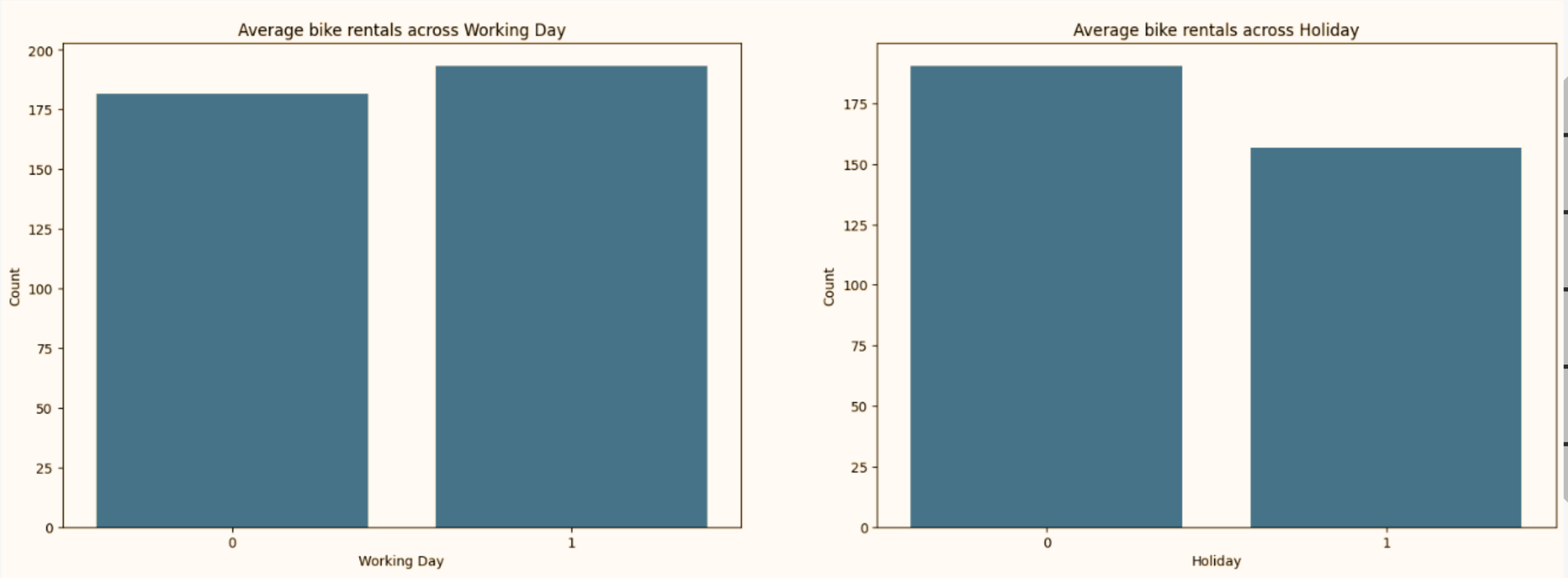


▶Highest bike reservations during Summer and Fall and lowest in Spring

▶In weekdays graph, we observe an average number of users throughout the week.

▶Average bike rentals are between 175 and 200 for weekdays

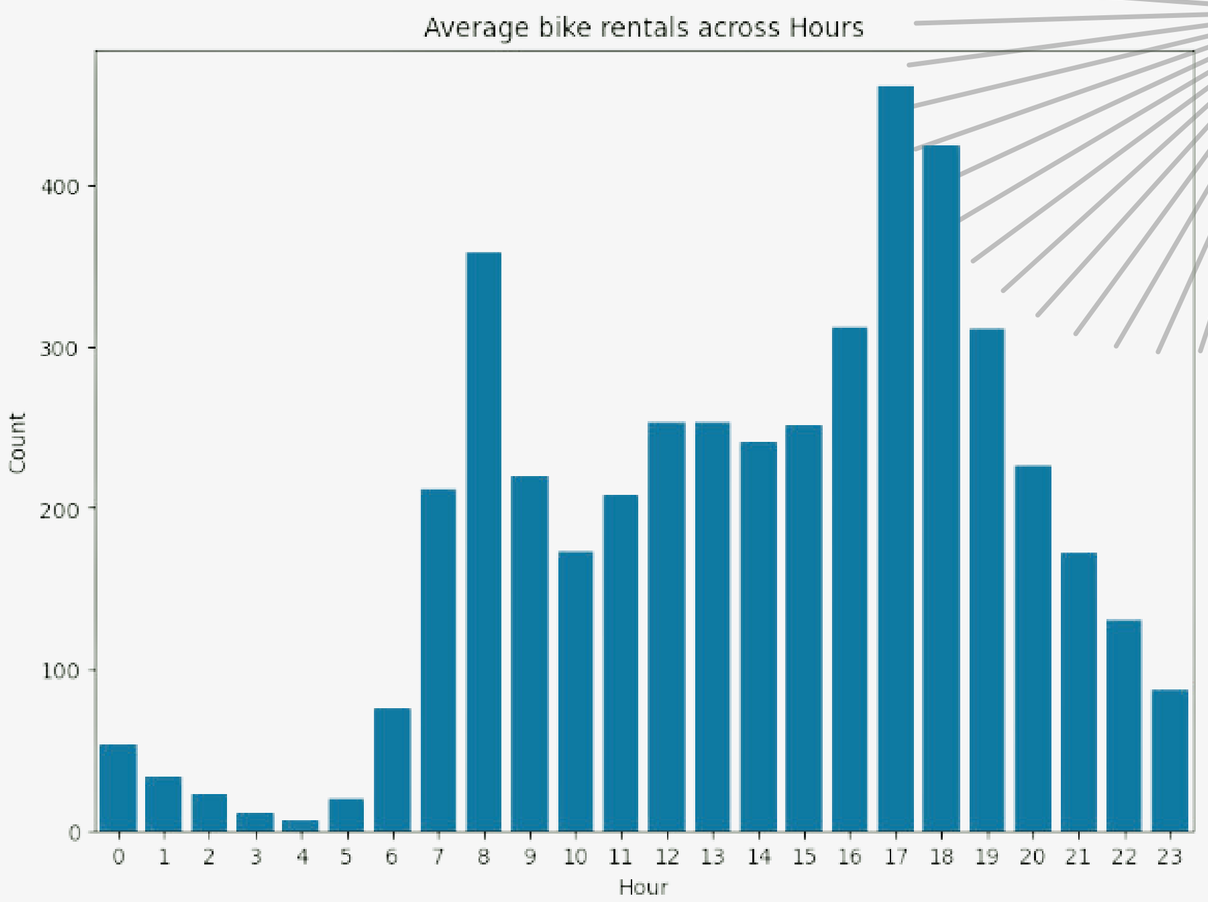
Holiday & Workingday



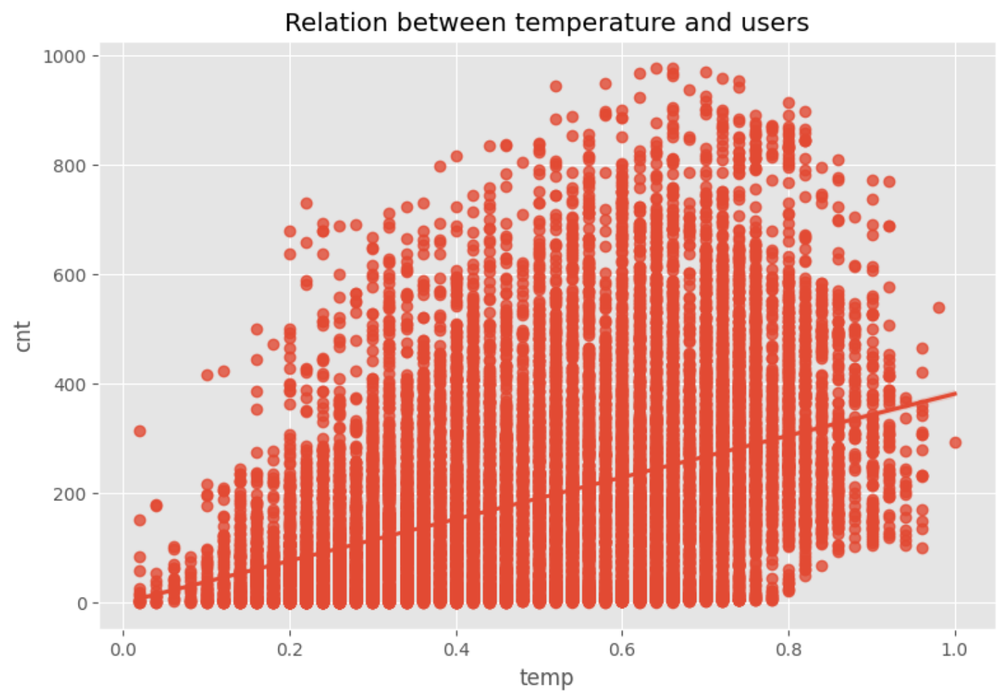
Average Bike rental counts for working day and Holidays.

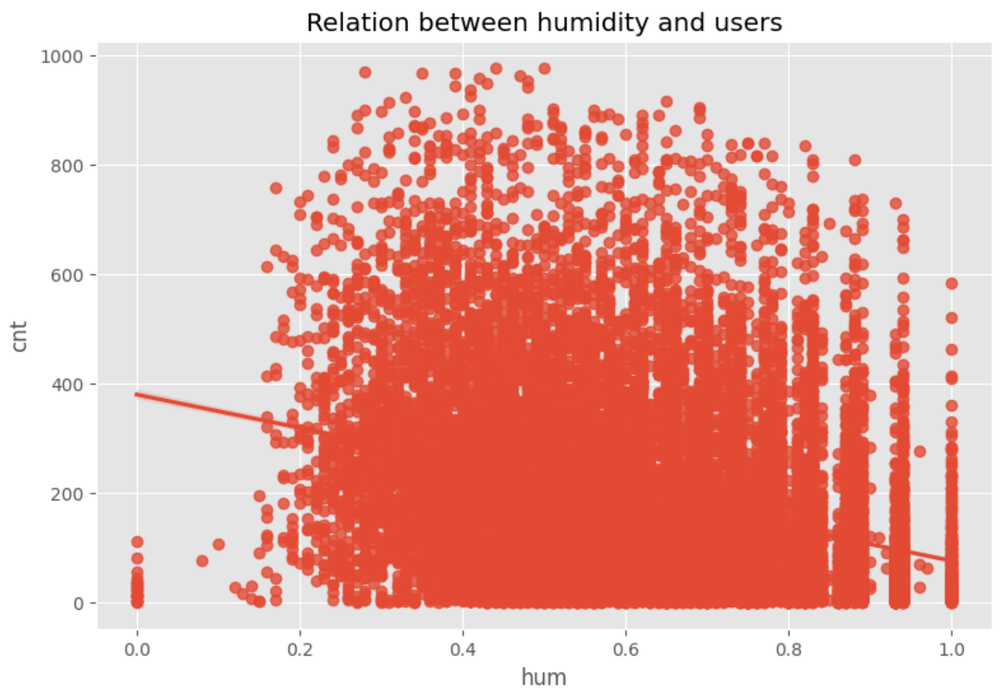
Hours

Most bike rentals are at 8 am and between 5 to 6 pm around the time when people travel to and fro to office or school and least rentals during early and late hours of a day.



Temperature/Humidity





With the increase in temperature, the number of users increases.

When the humidity increases the number of users renting bikes decreases.

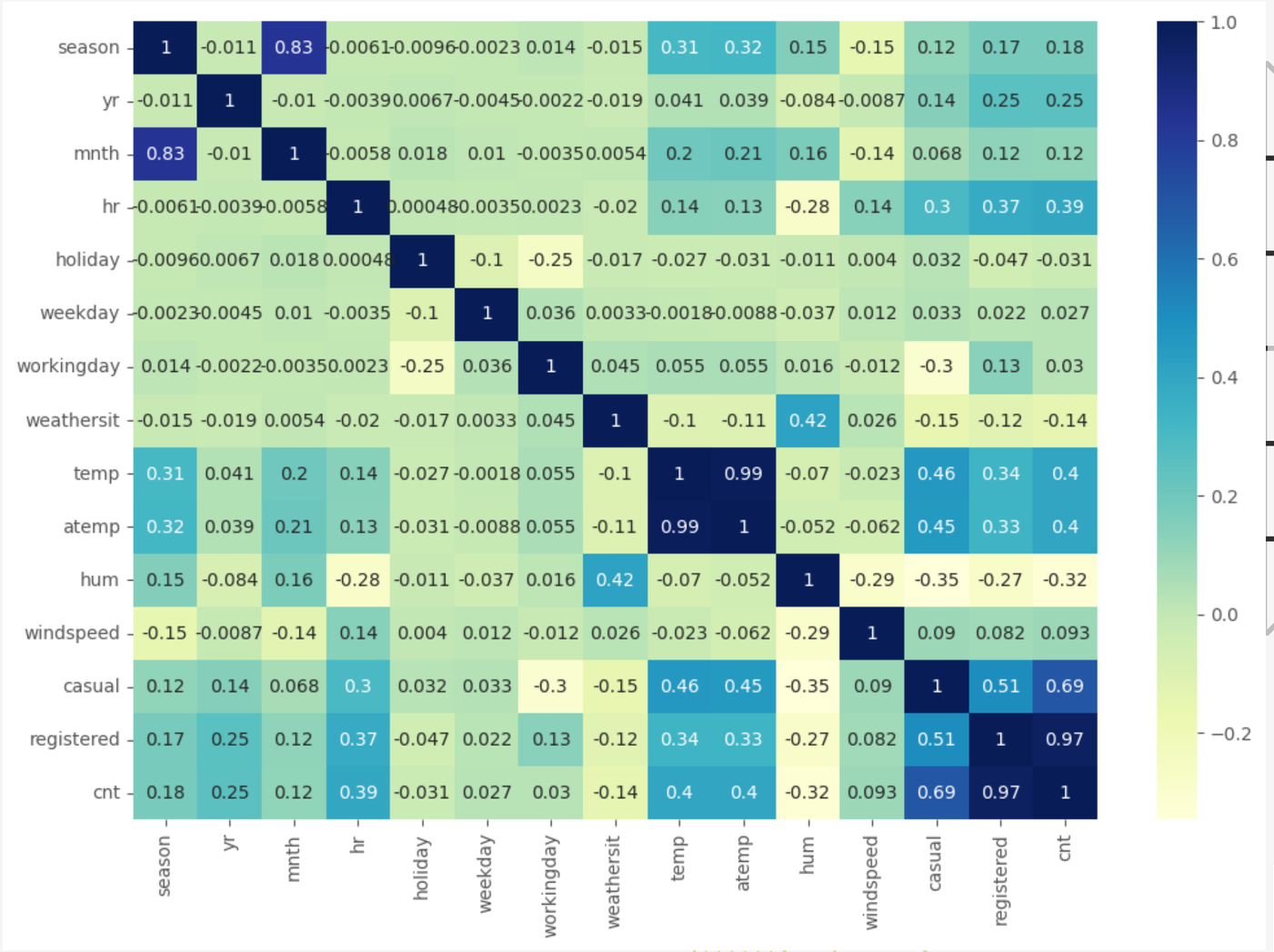
Data Preprocessing

- All the necessary libraries were imported first such as Numpy, Pandas, Matplotlib, Seaborn.
- Checking the basic profile of the dataset. To get a better understanding of the dataset.
  - Using Info method
  - Using Describe method
  - Checking for unique values of each column.
- Checking for null values, There are no null values present in our dataset.
- The instant column is just acting as an index column for us, we can drop that column.
- We already have month, weekday, year data as separate columns hence dteday is not required. Hence dropping it.
- We will also be dropping casual and registered column since the cnt column is already feature engineered with addition of casual and registered column in count column (casual+registered=cnt). Also it has been understood that increasing casual or registered users both will be profitable factor for the business.
- Next slide shows dropping of few other columns based on insights from Correlation heatmap

Correlation Analysis

▶temp and atemp are very highly corelated and their respective colinearities with cnt are also same. Hence dropping atemp since feeling temperature can be relatively less accurate compared to temperature.

▶ Dropping holiday column as it is highly correlated to'workingday column.



Remaining Data Preprocessing

Have converted the integer columns season, yr, mnth, hr, holiday, weekday, workingday, weathersit into categorical columns.

● So, finally following are the predictor variable that will be used for prediction

numerical_columns = ["temp", "hum","windspeed"]

categorical_columns = ["yr","mnth","hr","weekday","season","workingday","weathersit"]

Standard Scaling has been done on Numerical and Categorical columns.

One Hot Encoding has been done on Categorical columns

Model Workflow (Model Selection)

The data was split into 2 sets X and y. X contains all the columns except the target column in our case (Count), y contains only the Target column.

- Using train test split we first split the dataset into X_train,X_test, y_train, y_test .
- Following Regression Algorithms were used: CatBoost, Random Forest, Bagging, Decision Tree, Gradient Boost, KNeighbors, Linear Regression.
- After creating the following normal without any hyper tuned models we get CatBoost Regressor having the highest accuracy of 93.35 on testing data.

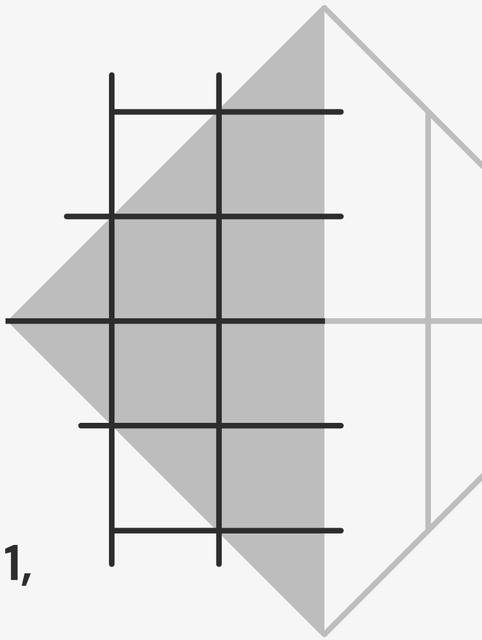
Model Workflow (Model Accuracy Score)

● We used Grid Search CV to find the best suiting parameters for our Catboost Regressor model. Grid Search CV was used on the following set of parameters.

```
parameters={
    "iterations": [1000],
    "learning_rate": [1e-3, 0.1],
    "depth": [1, 10],
    "subsample": [0.05, 1.0],
    "colsample_bylevel": [0.05, 1.0],
    "min_data_in_leaf": [1, 100]

}
```

Hyperparameter Tuning on Best Model
Cat Boost Regressor



•Best Hyperparameters Found for this Best Model

•colsample_bylevel: is the subsample ratio of columns for each level= 1.0,
•depth: Depth of tree= 10,
•iterations: number of iterations= 1000,
•learning_rate: this setting used for reducing gradient step= 0.1,
•min_data_in_leaf: parameter to prevent over-fitting in a leaf-wise tree = 1,
•subsample: Subsample ratio of the training instance= 1.0

These best parameters are passed into the CatBoost Regressor model thus creating a Hyper Parameter tuned model.

Actual Vs Predicted

Table at right shows that there is very less difference between Actual value and Predicted value got from HypertunedCatBoost Regressor model.

Training Accuracy: 98.8%

Testing Accuracy: 94.4%

	Actual Value	Predicted Value	Difference
0	425.0	395.710051	29.289949
1	88.0	90.620582	-2.620582
2	4.0	12.302707	-8.302707
3	526.0	516.357435	9.642565
4	13.0	14.096443	-1.096443
...	...	...	...
3471	17.0	18.974234	-1.974234
3472	85.0	83.613447	1.386553
3473	98.0	74.182476	23.817524
3474	266.0	303.229539	-37.229539
3475	267.0	237.883117	29.116883

Comparing Error rate Before and After Hypertuning our CatBoost model

Before hyperparameter tuning:

```
MAE: 30.29550950931159
MSE: 2149.264506521926
RMSE: 46.36016076893959
```

After hyperparameter tuning:

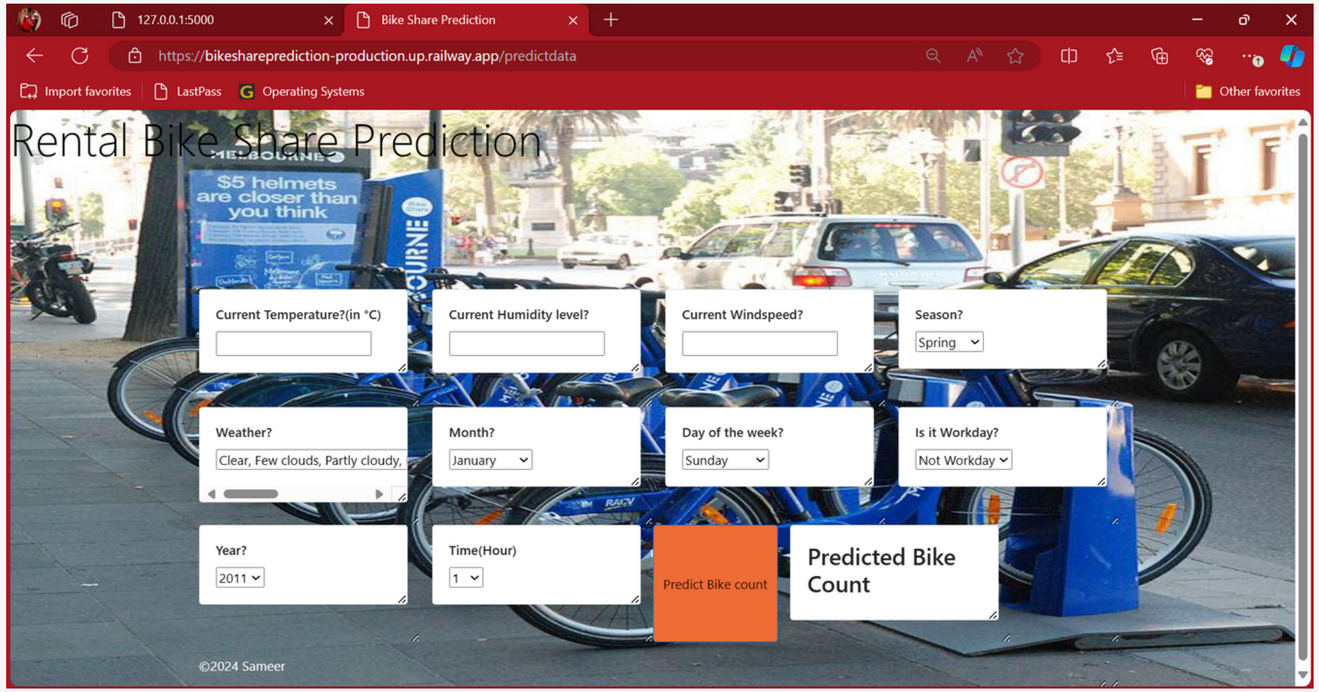
```
MAE: 26.86409951098349
MSE: 1788.167274967652
RMSE: 42.28672693609251
```

The model was pickled using the Python pickle library and was ready for use into our Backend system.

Data Collection (Input from User)

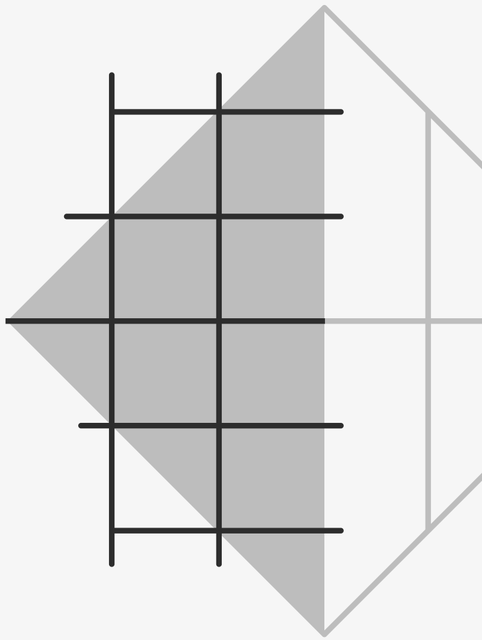
The data from the user is being retrieved from the front-end web page application which is created using HTML , CSS & Java Script. Where users input as shown in the image in next slide.

URL for this web app is:
https://bikesharepredication-production.up.railway.app/predictdata



Data Validation process

The data is taken from the frontend web app then sent to the machine learning model but, before that data which is entered by the user and its been validated by the app.py we created as a Flask application.



Rendering the result

The data which is taken as the input from the user is sent to the machine learning model and then the result is rendered on the front-end HTML page.

Application Deployment

This model is deployed on railway.app (a platform like Heroku). The following are the steps to deploy this application:

Follow the steps to deploy on railway app using any of the instruction videos available on YouTube.

Instruction video link:
https://www.youtube.com/watch?v=WzxEHzb2NzM

Conclusions

•Out of the 7 models tried, CatBoost Regressor model yielded best prediction accuracy of 94.4% on Test Data

•We have successfully built an end-to-end web application using machine learning that can help predict the total count of bikes rented by the users based on various conditions.

•This type of system can help users to get a better understanding of whether the day is good for riding bikes and make their decision making easier.

•This system will help companies to help the users and provide a better end to end user experience, increase efficiency, fulfill bike rental demands,avoid wastage of resources by only deploying required number of bikes thus saving cost and space for parking.

Q&A

Q1) What's the source of data?

The dataset was taken from UCI Machine learning repository

https://archive.ics.uci.edu/dataset/275/bike+sharing+dataset

Q 2) What was the type of data?

The data was the combination of numerical and Categorical values.

Q 3) What's the complete flow you followed in this Project?

Please refer to slide 10 for better Understanding

Q 4) What were the libraries that you used in Python?

I used libraries like Pandas, Numpy, Matplotlib, Seaborn, Scikit-Learn.

Q 5) How logs are managed?

We are mentioning logging information in between the code as it proceeds, the logs are created for steps like Data Ingestion, Train-test split, Obtaining & saving preprocessing object, Names of Categorical & Numerical columns used for prediction, Model Training, Model evaluation, Hyperparameter tuning, Finding best model, etc.

Q 6) What techniques were you using for data pre-processing?

- Removing unwanted attributes
- Visualizing relation of independent variables with each other and output variables
- Checking for outliers
- Cleaning data.
- Converting certain numeric data to categorical data.
- Scaling the data

Also refer slides 17, 18, 19 for more details and understanding.

Q 7) How training was done or what models were used?

▶Refer slides 20 to 24 for details

▶ Regression Algorithms like Catboost, Bagging, Random Forest, Decision Tree, Gradient Boost, KNeighbors, Linear Regression were used and the best performing model out of all was CatBoost Regression model.

Q 8) Deployment?

Refer slide 27 for details

Thank you