

# **CAPSTONE PROJECT**

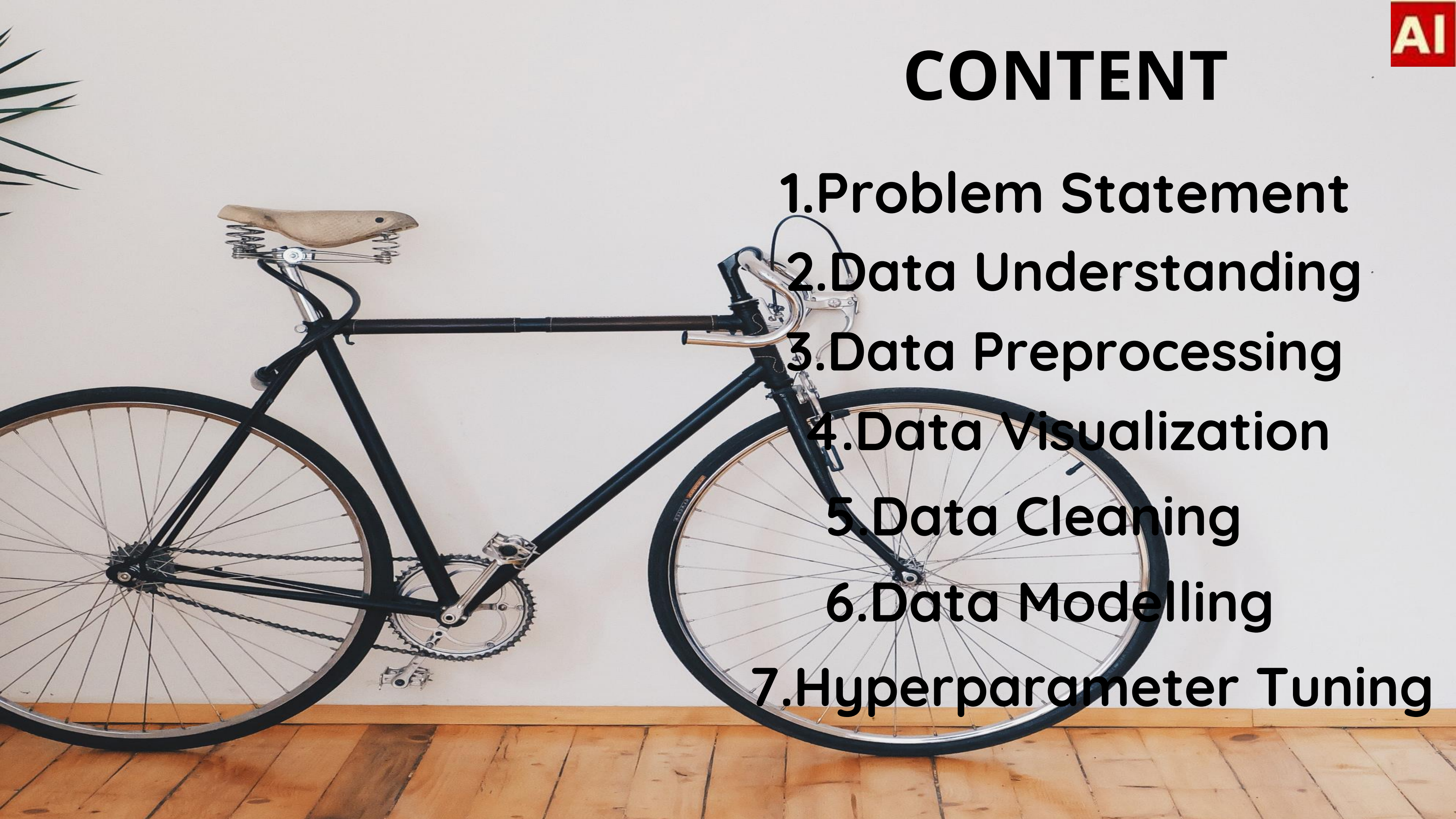
## **SEOUL BIKE SHARING DEMAND PREDICTION**

### **SUPERVISED LEARNING (REGRESSION)**

**Presented By:**  
**RAJ SONAR**



# CONTENT

- 
1. Problem Statement
  2. Data Understanding
  3. Data Preprocessing
  4. Data Visualization
  5. Data Cleaning
  6. Data Modelling
  7. Hyperparameter Tuning



# PROBLEM STATEMENT

- Currently Rental bikes are introduced in many urban cities for the enhancement of mobility comfort. It is important to make the rental bike available and accessible to the public at the right time as it lessens the waiting time. Eventually, providing the city with a stable supply of rental bikes becomes a major concern.
- The crucial part is the prediction of bike count required at each hour for the stable supply of rental bikes.

The main goal of the project is to find factors that cause and influence shortages of bike and time delay for availing bike on rent. Using the data provided, this presentation aims to analyze the data to determine what variables are correlated with bike demand prediction. Hourly count of bike for rent will also be predicted.

# Columns Description



## Attribute Information

- **Date:** Day-Month-Year Format
- **Rented Bike Count:** Count of bikes rented at each hour
- **Hour:** Hour of the day
- **Temperature:** Temperature of the hour in Celsius
- **Humidity:** Humidity% of the hour
- **Windspeed:** Wind Speed of the hour in m/s
- **Visibility:** Visibility of the hour in units of 10m
- **Dew Point Temperature:** Dew Point Temperature of the hour in Celsius
- **Solar Radiation:** Solar Radiation of the hour in MJ/m<sup>2</sup>
- **Rainfall:** Rainfall of the hour in mm
- **Snowfall:** Snowfall of the hour in cm
- **Seasons:** Winter, Spring, Summer, Autumn
- **Holiday:** Holiday/No holiday
- **Functional Day:** Non Functional Hours/Functional Hours

# Feature Breakdown

- **Date:** The date of the day, during 365 days from 01/12/2017 to 30/11/2018, formatting in DD/MM/YYYY (object type), we converted it into date-time format.
- **Rented Bike Count:** Number of bikes rented per hour which is our dependent variable and we need to predict that.
- **Hour:** The hour of the day, starting from 0-23 it's in a 24 hour format.
- **Temperature (°C):** Temperature of the hour in Celsius and it varies from -17°C to 39.4°C.
- **Humidity (%):** Availability of Humidity in the air during the hour and it ranges from 0 to 98%.
- **Wind Speed (m/s):** Speed of the wind at the renting hour and it ranges from 0 to 7.4m/s.
- **Visibility (10m):** Visibility to the eyes during the hour in “m” and it ranges from 27m to 2000m.
- **Dew point temperature (°C):** Temperature at the beginning of the day and it ranges from 0.6°C to 27.2°C.

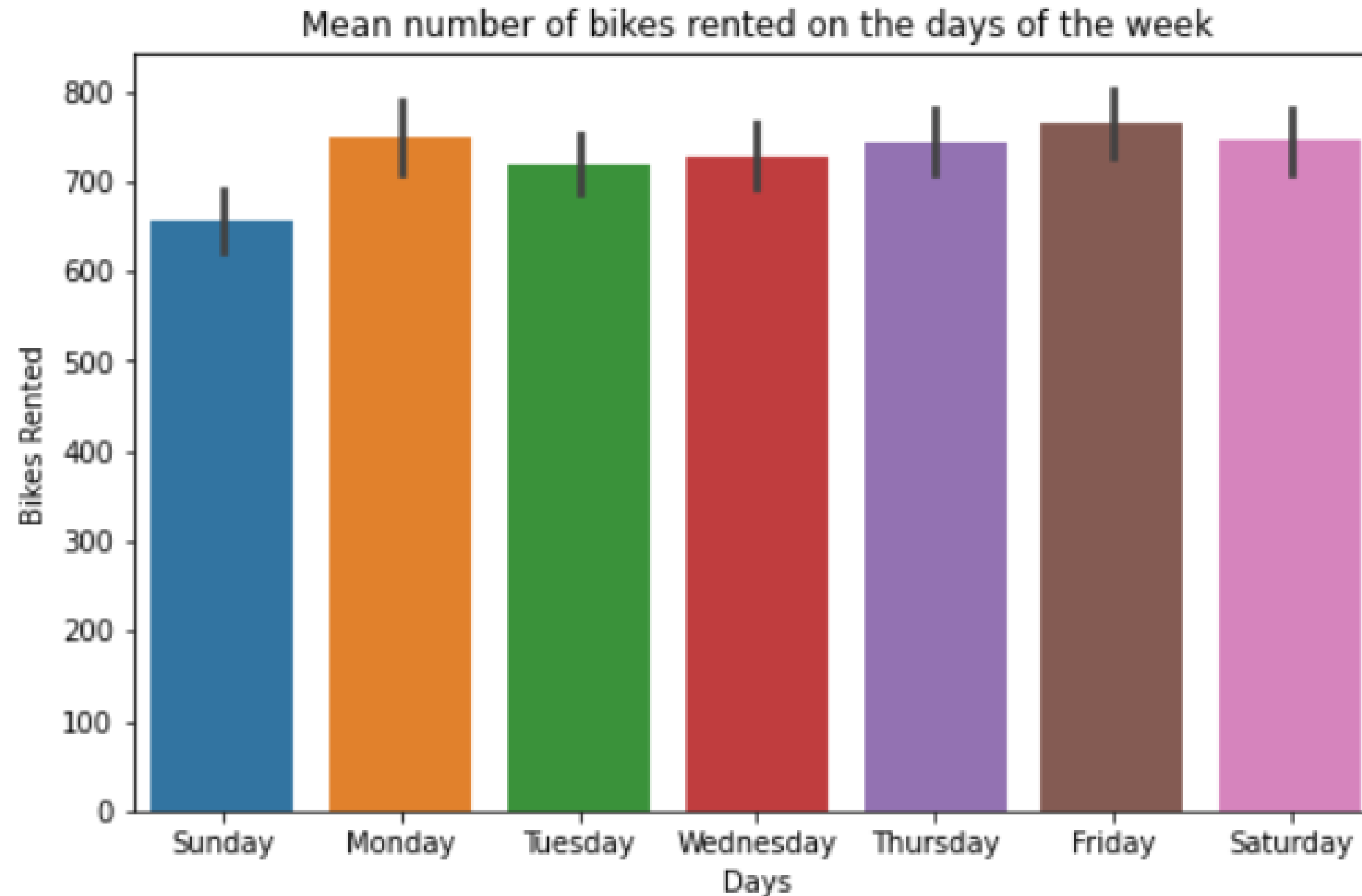
- **Solar Radiation (MJ/m<sup>2</sup>):** Sun contribution or solar radiation during the hour of ride booking which varies from 0 to 3.5 MJ/m<sup>2</sup>.
- **Rainfall (mm):** The amount of rainfall during bike booking hour which ranges from 0 to 35mm.
- **Snowfall (cm):** Amount of snowfall in cm during the booking hour in cm and ranges from 0 to 8.8 cm.
- **Seasons:** Seasons of the year and total there are 4 distinct seasons i.e. summer, autumn, spring and winter.
- **Holiday:** If the day is holiday period or not and there are 2 types of data that is holiday and no holiday
- **Functioning Day:** If the day is a Functioning Day or not and it contains object data type yes and no.

# Data Preprocessing

- **Checked the shape of the dataset where the rented bikes count was 0 and then checked the shape of the dataset where the store was not functional and found out that the shape of both the datasets is the same and it was quite obvious that 0 bikes would have been rented when the store was not functional. So removed the rows where the count of rented bikes was equal to 0.**
- **Converted the 'Date' column to date-time datatype and extracted the year, month and the day from the 'Date' column for data analysis.**

# Exploratory Data Analysis

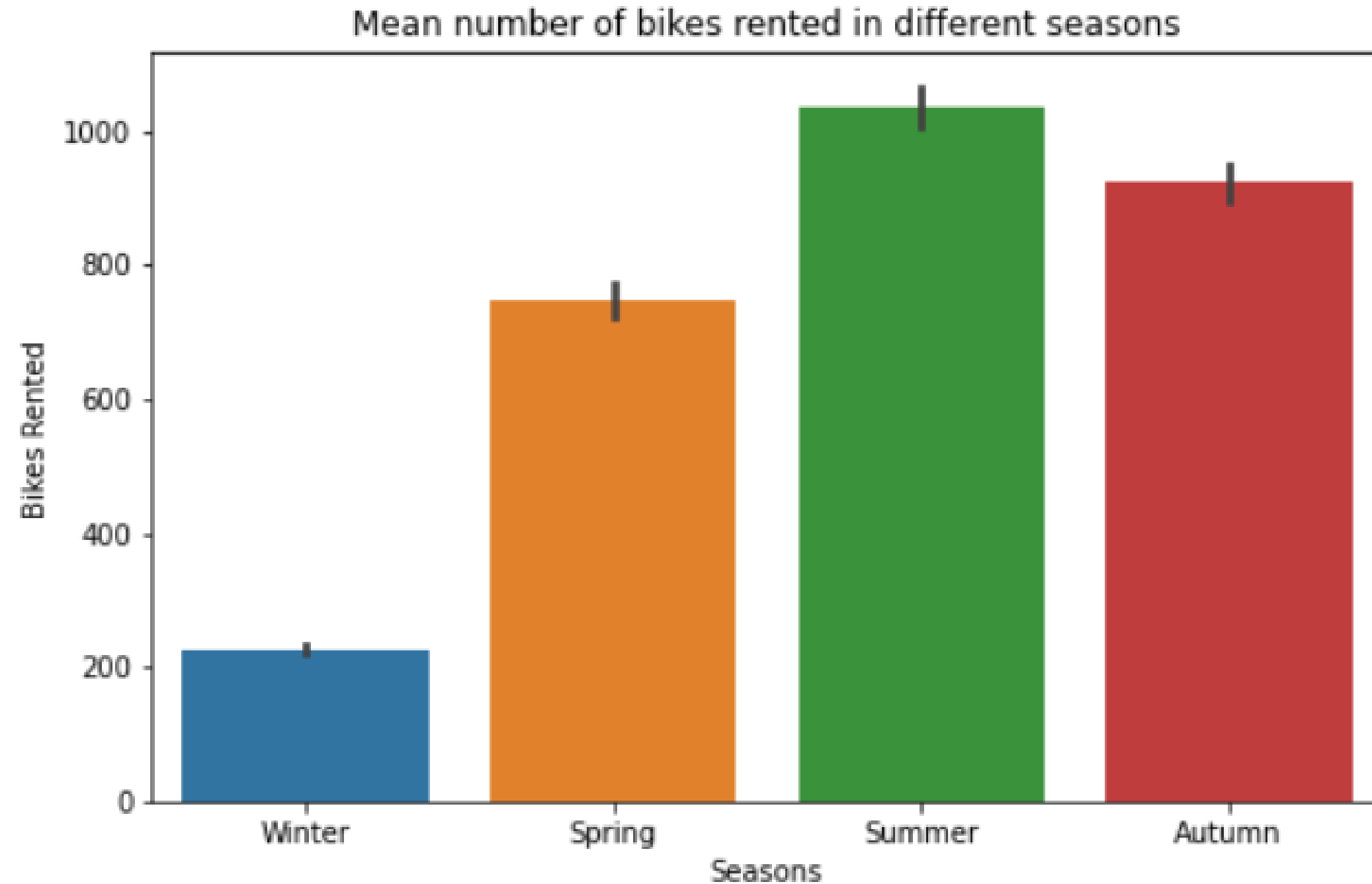
## Mean number of bikes rented on the days of the week



**Least numbers of bikes are being rented on Sundays, it shows people tend to rent the bikes for work and office purposes**

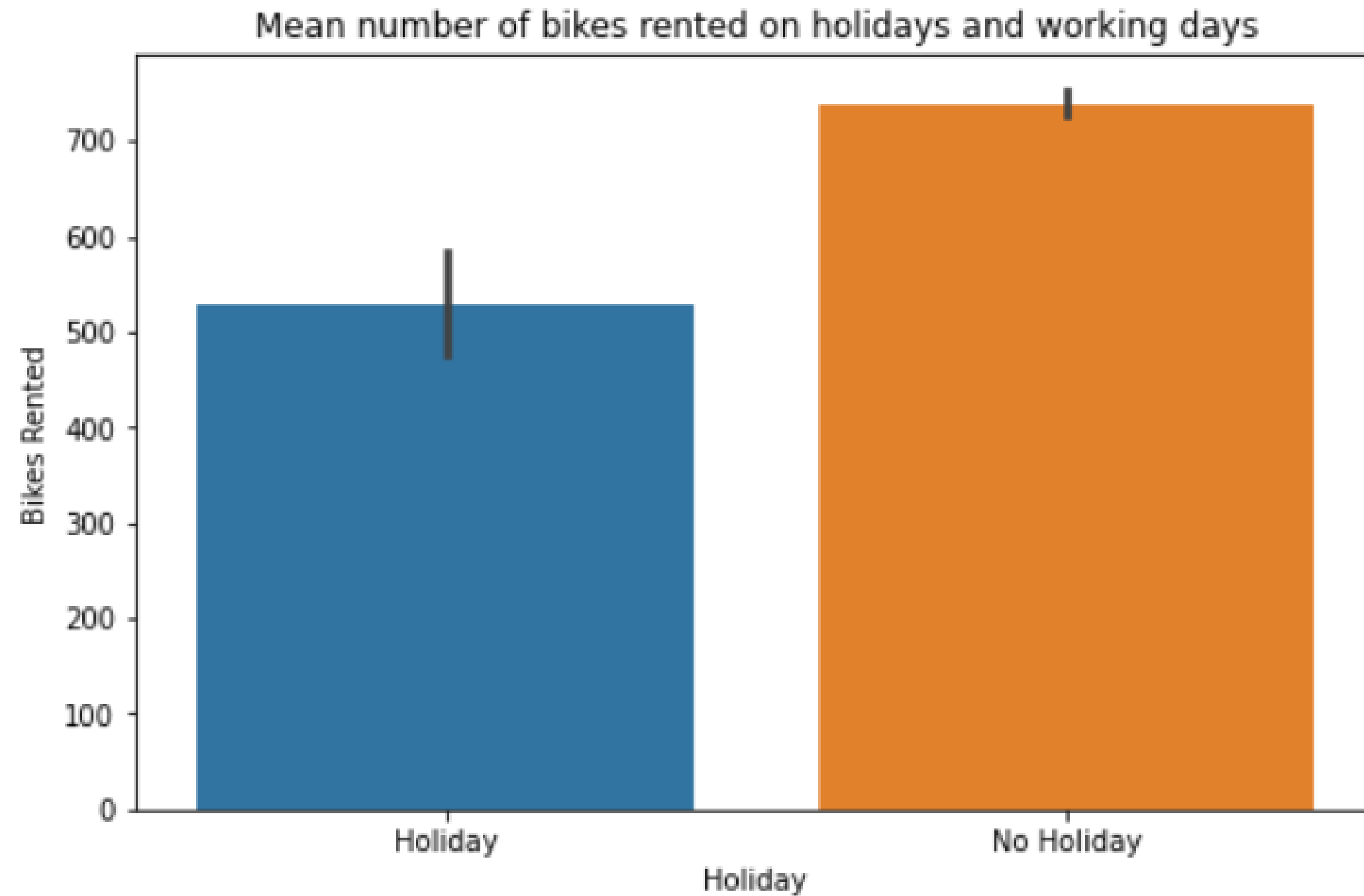


## Mean number of bikes rented in different seasons



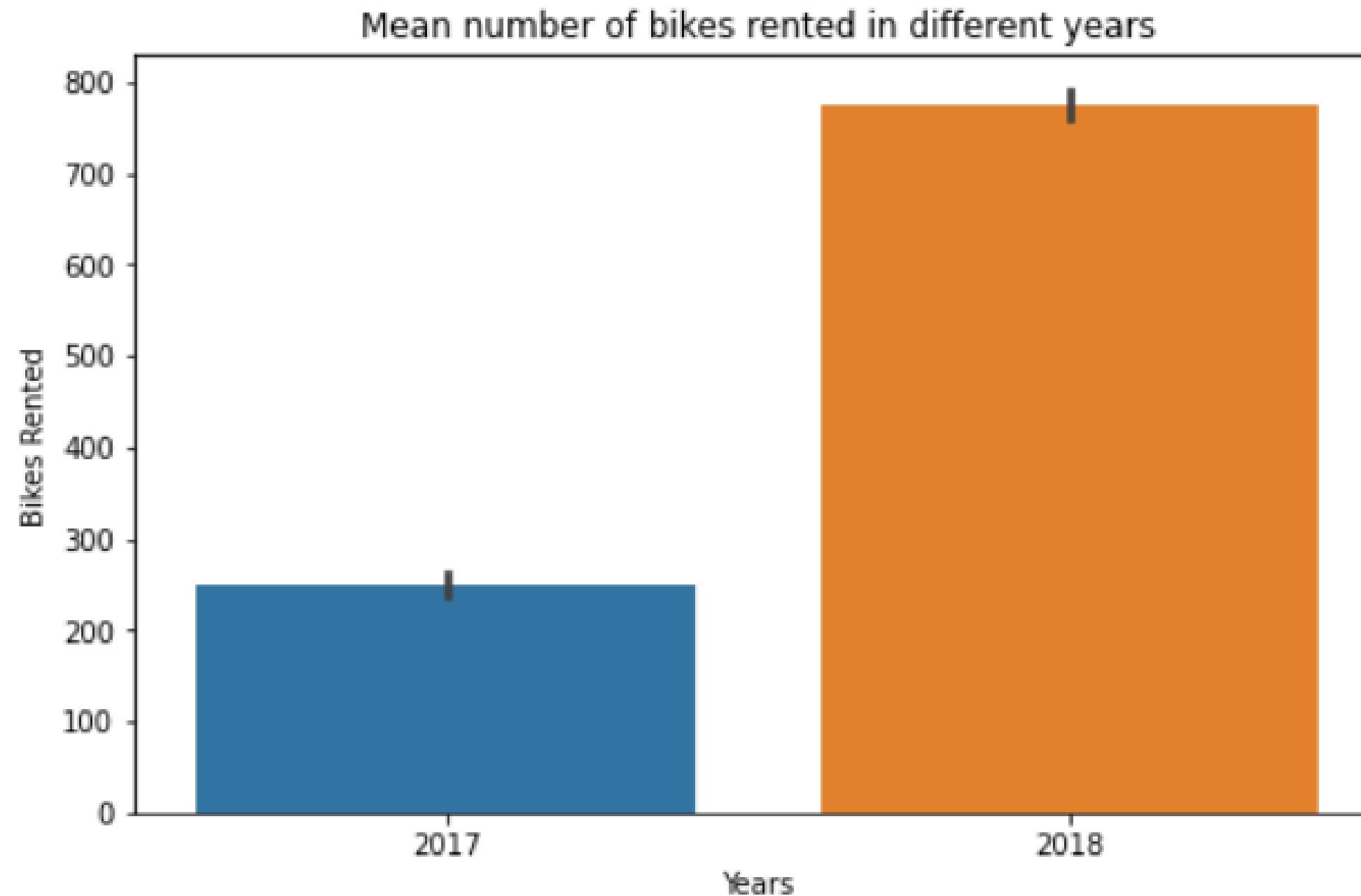
**Maximum bikes are being rented in the Summer Season and least number of bikes are being rented in the Winter Season**

## Mean number of bikes rented on holidays and working days



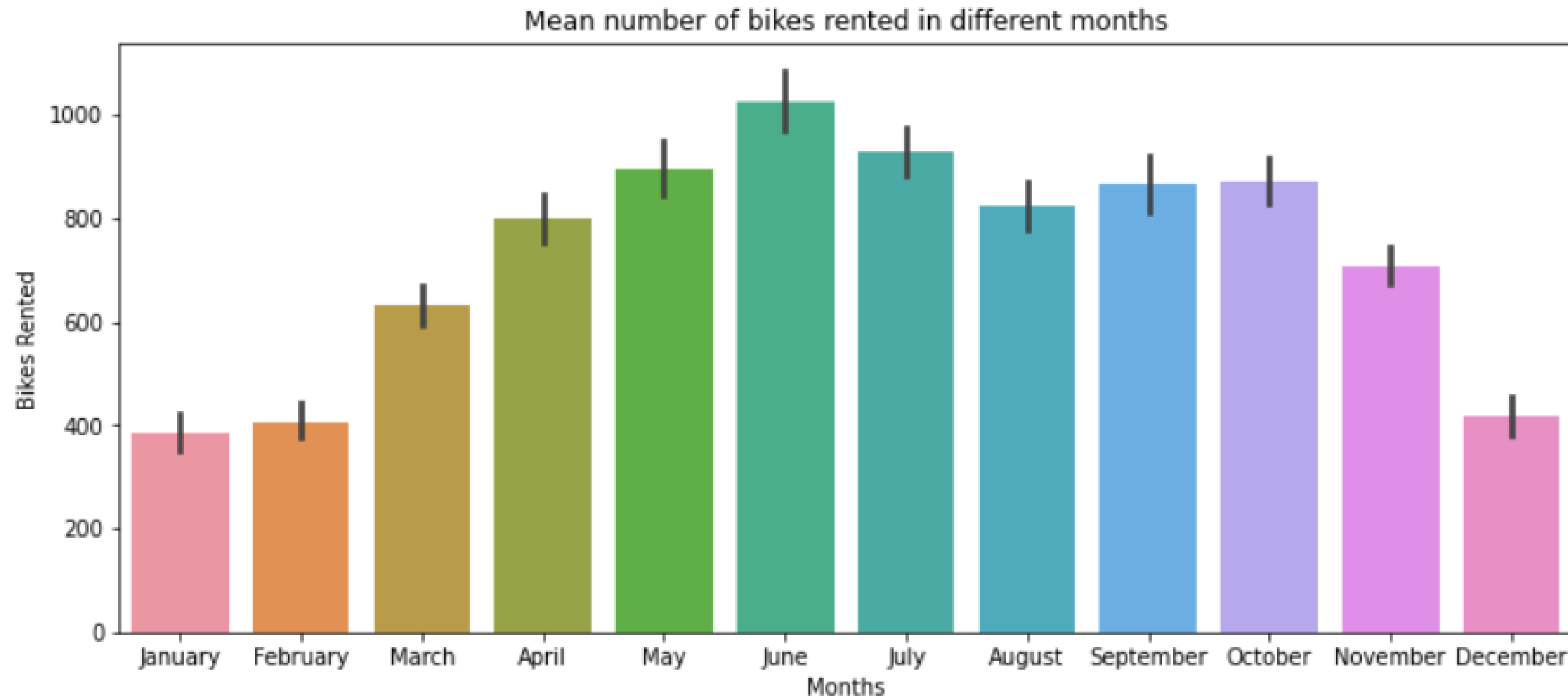
**More number of bikes are being rented on the working days indicating that people rent bikes for work purposes more than leisure purposes**

## Mean number of bikes rented in different years



**The graph shows that the bike renting business is a growing business as the number of bikes rented in 2018 is more than the bikes rented in 2017**

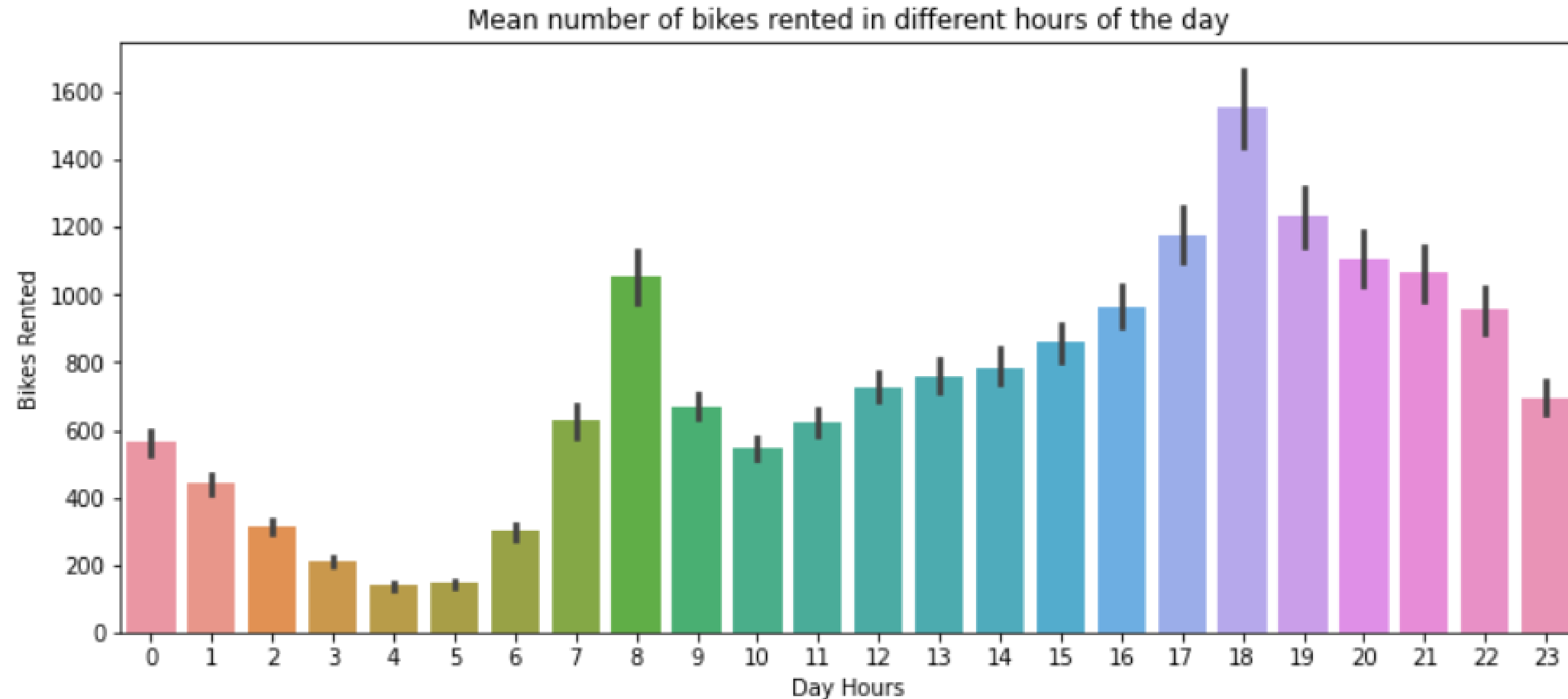
## Mean number of bikes rented in different months



**Maximum number of bikes are getting rented in the month of May, June and July which are the summer months and it can be seen that people love riding the bikes in the summer season**



## Average number of bikes rented in different hours of the day



- Maximum number of bikes being rented in the evening time are at 5pm, 6pm and 7pm showing that people mostly rent bikes in the evening for leisure purposes after the office time.
- Maximum number of bikes being rented in the morning time are at 8am showing that people rent bikes for going to office.

# CORRELATION HEATMAP

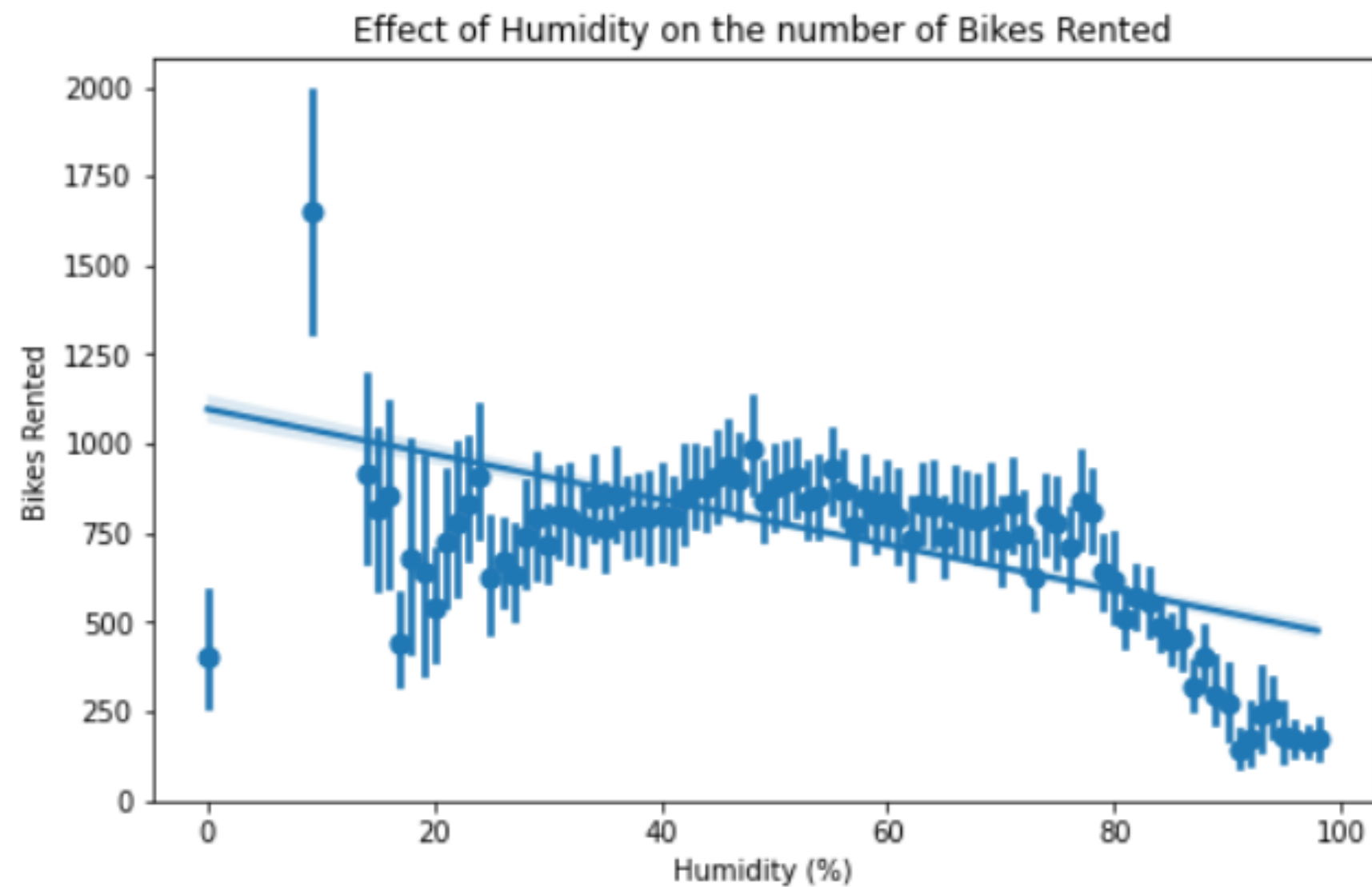


Maximum collinearity between dependent and independent variable:

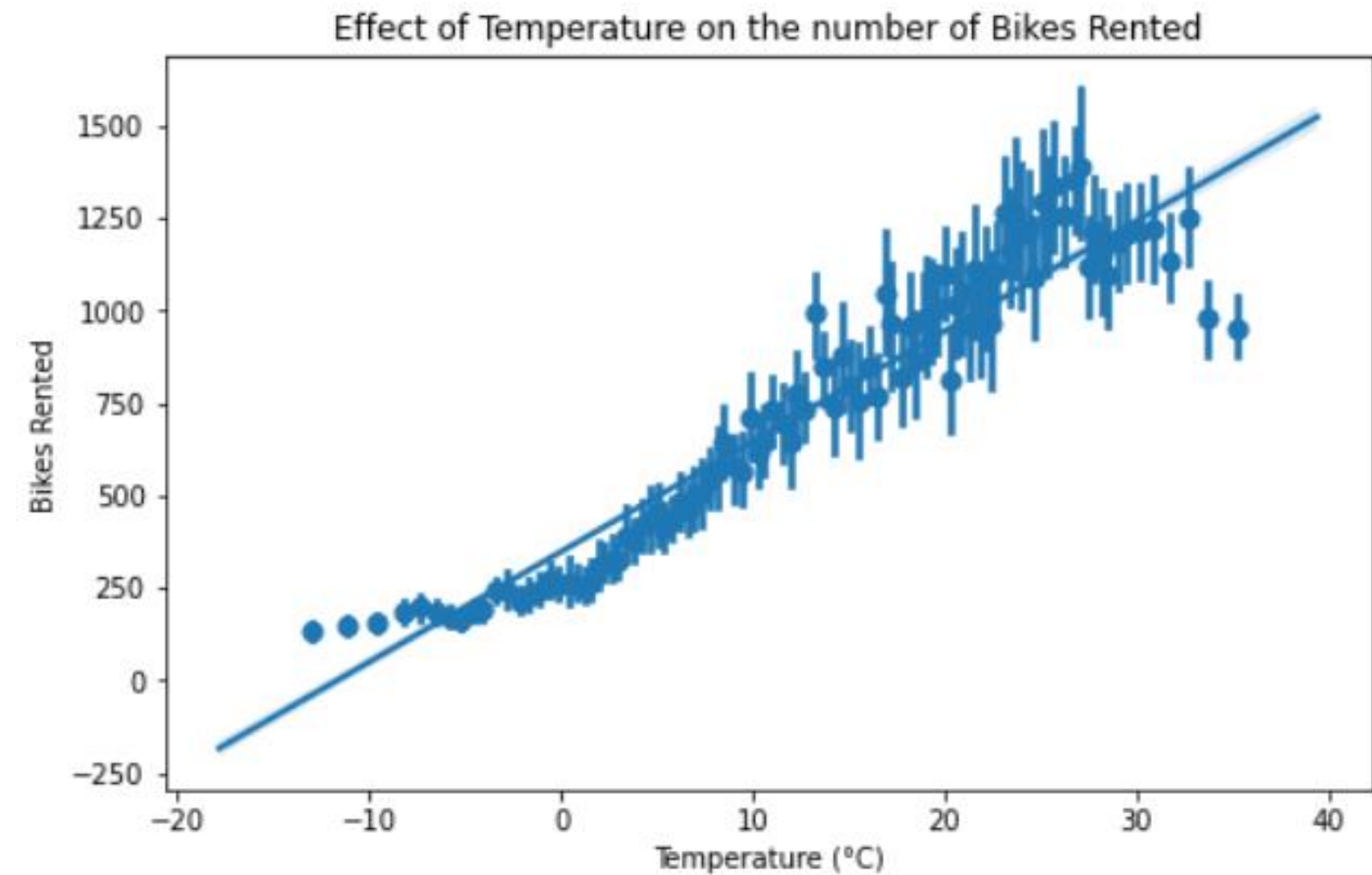
- Temperature and Rented Bike Count
- Hour and Rented Bike Count

Maximum collinearity between independent variables (Multi-Collinearity):

- Temperature and Dew Point Temperature
- Visibility and Humidity
- Dew Point Temperature and Humidity
- Solar Radiation and Humidity

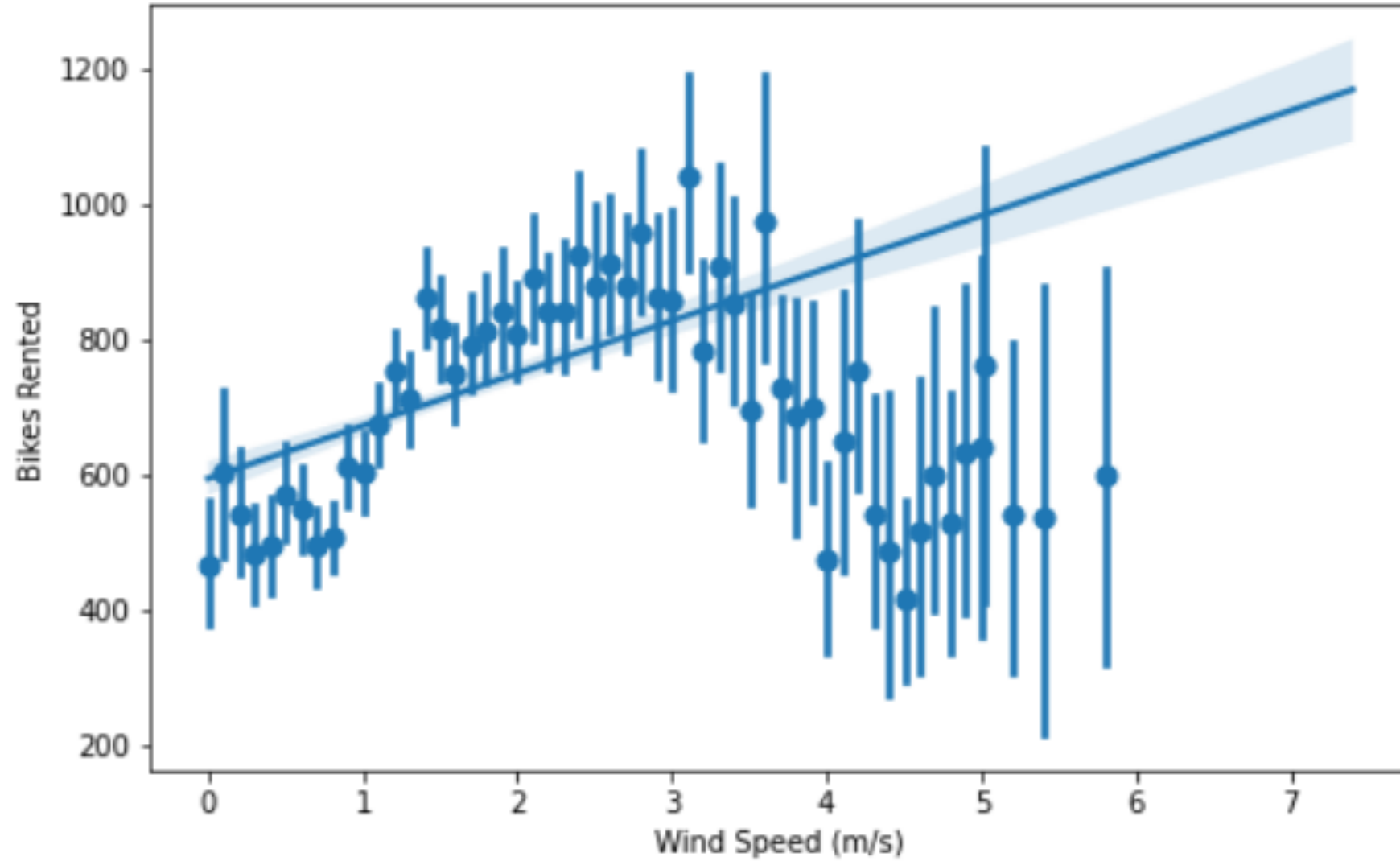


**Number of bikes rented and Humidity shows a negative correlation**



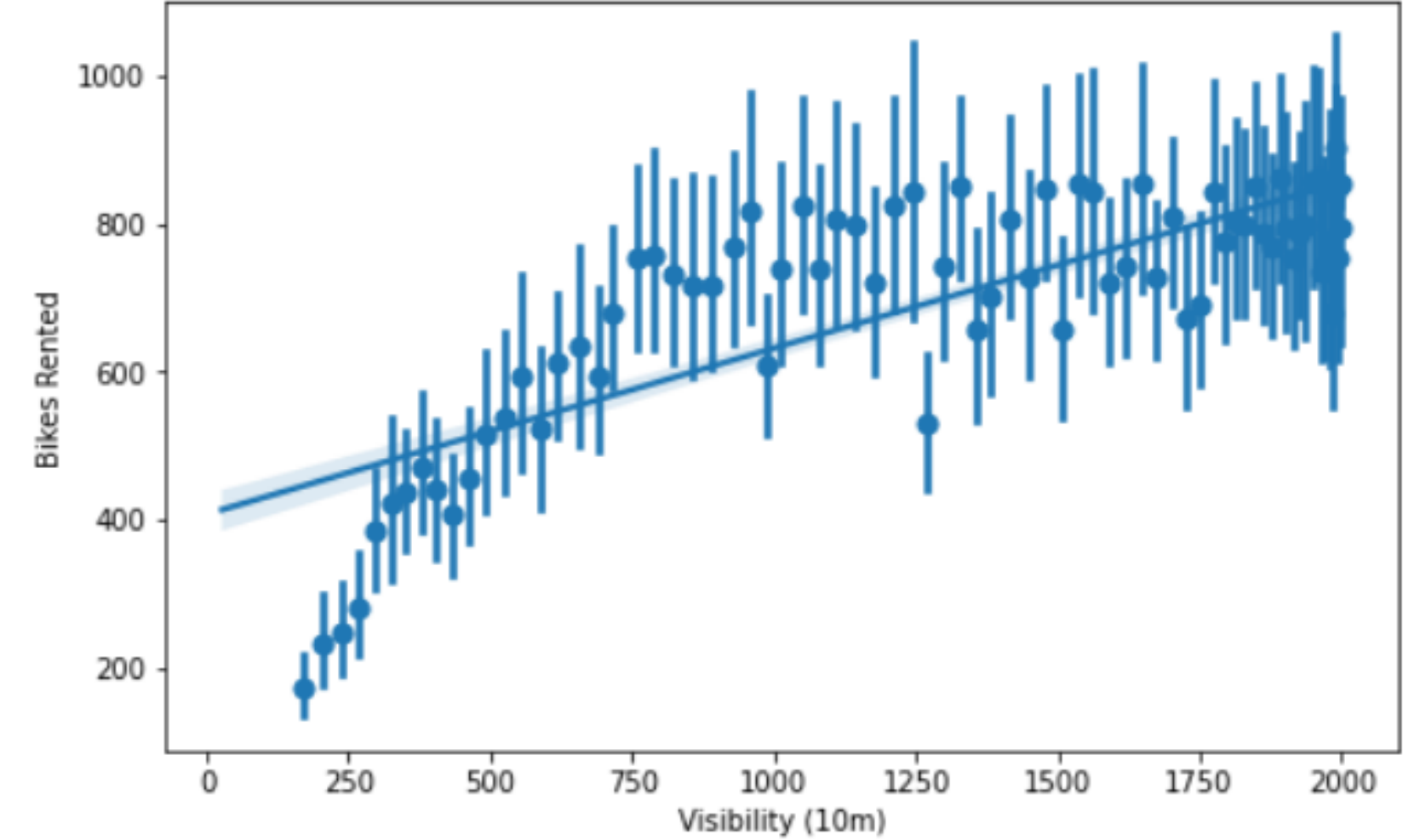
**Number of bikes rented and Temperature shows a positive correlation**

Effect of Wind Speed on the number of Bikes Rented



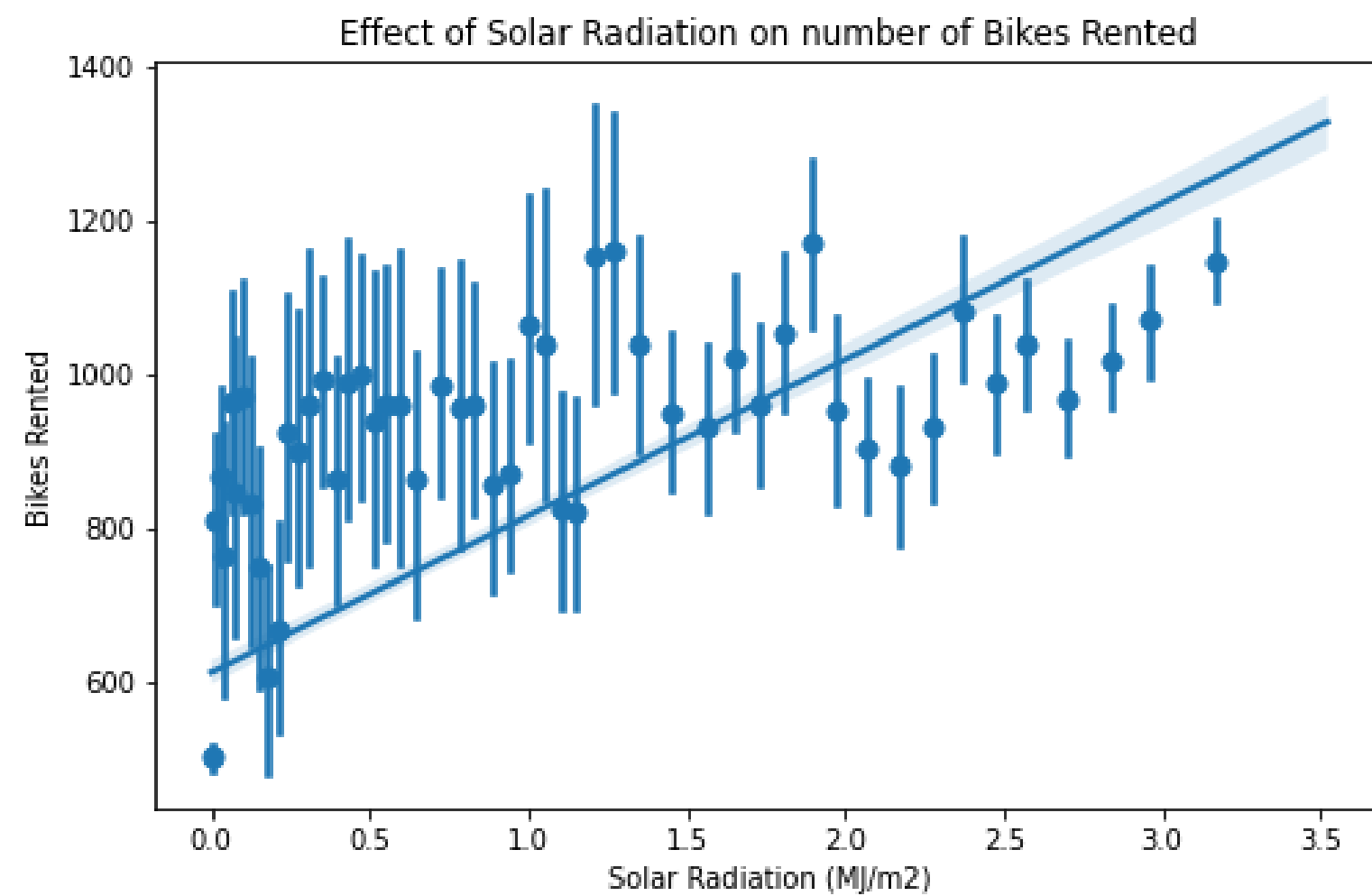
**Number of bikes rented and Windspeed  
does not show a good correlation**

Effect of Visibility on the number of Bikes Rented

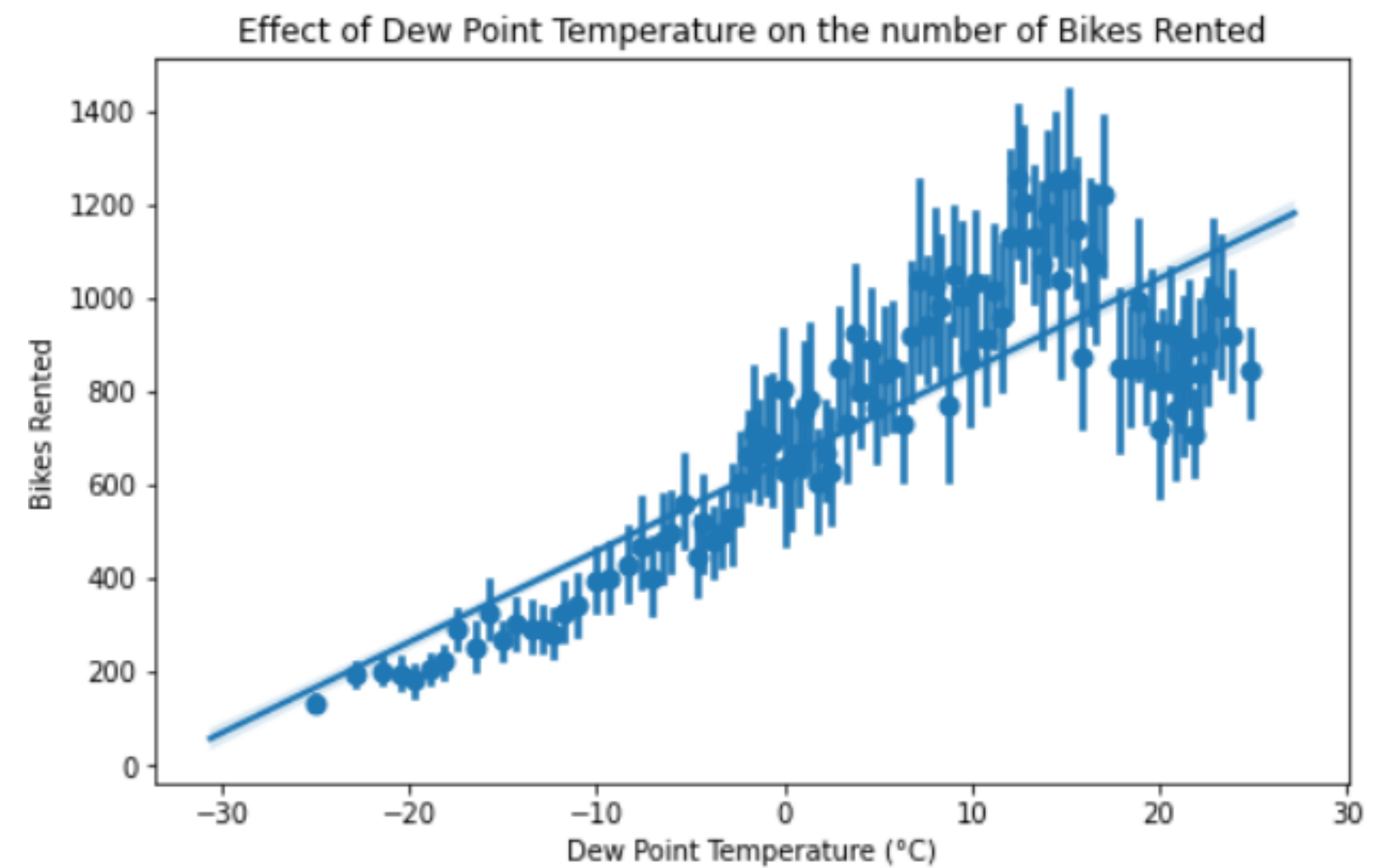


**Number of bikes rented and Visibilty  
shows a positive correlation**

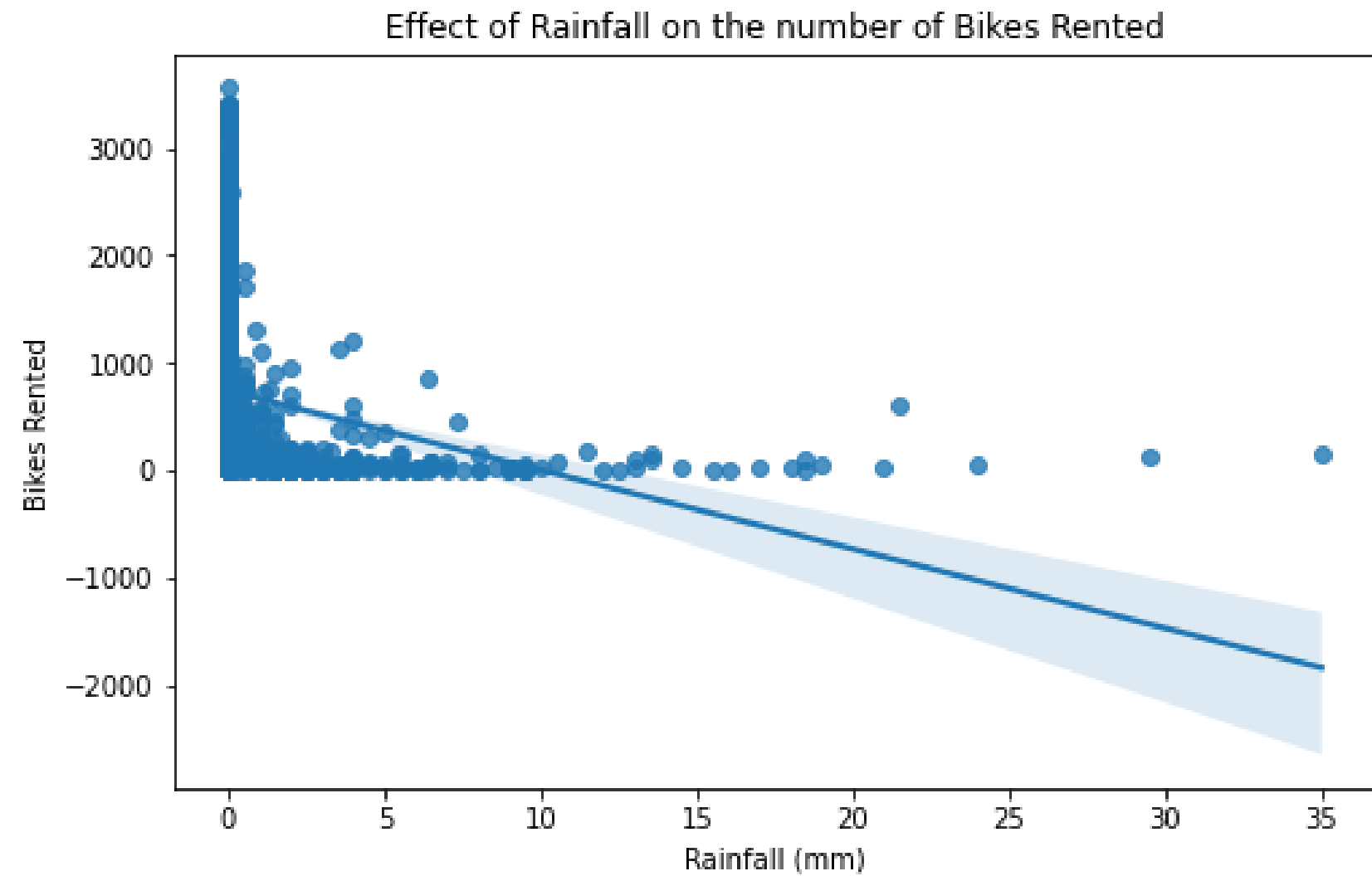




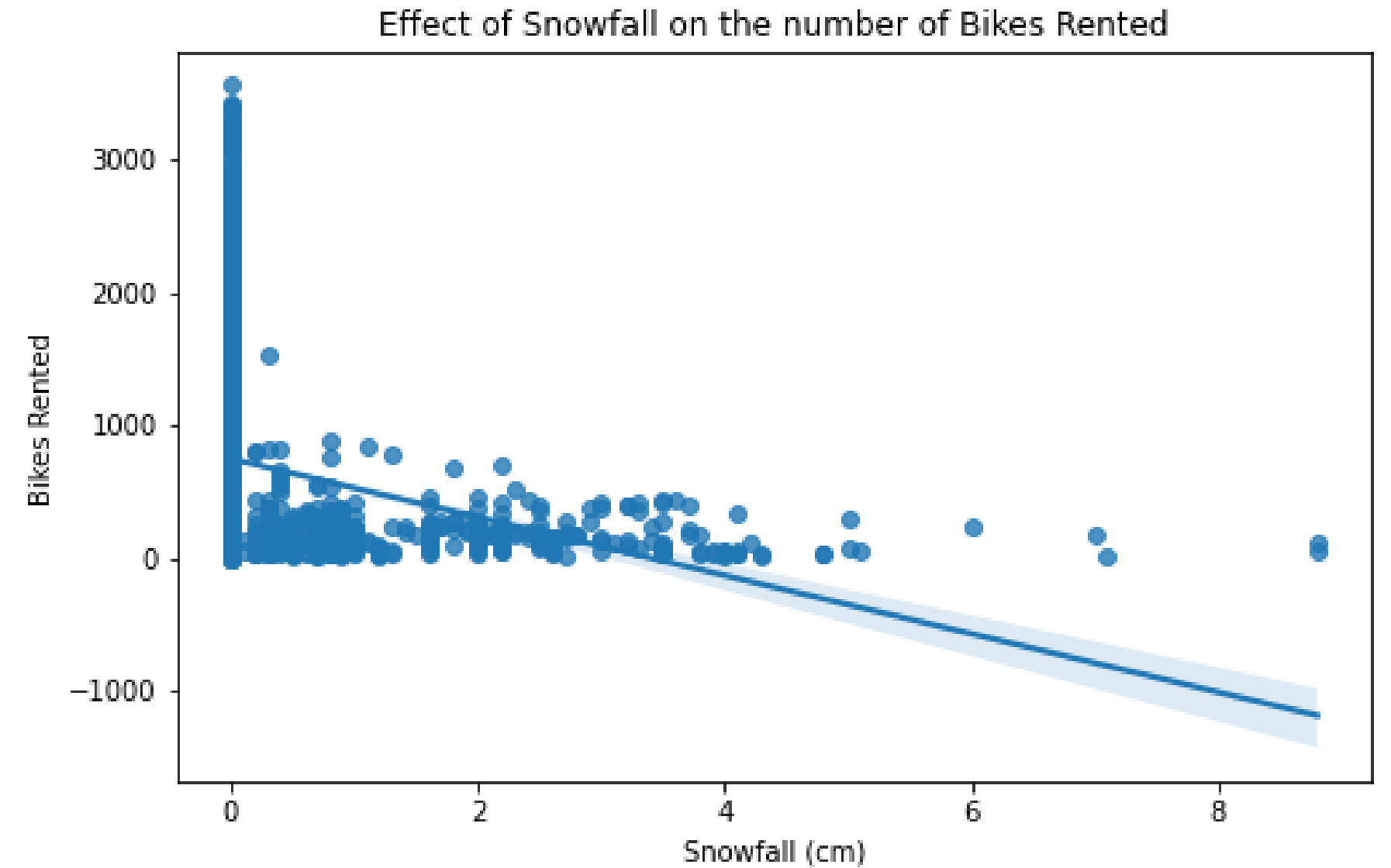
**Number of bikes rented and Solar Radiation shows a positive correlation**



**Number of bikes rented and Dew Point Temperature shows a positive correlation**

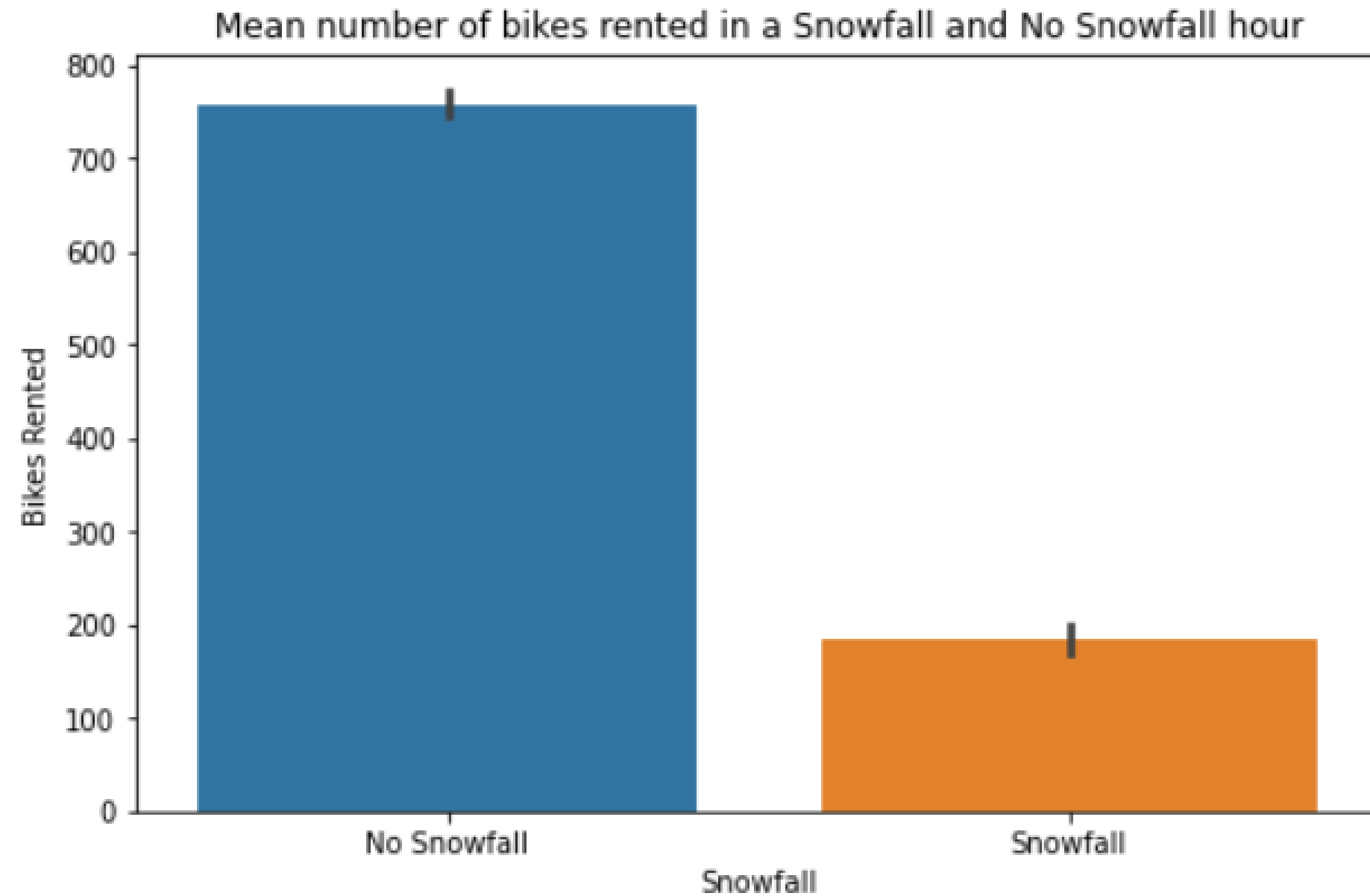


**Number of bikes rented and Rainfall  
does not show a good correlation**

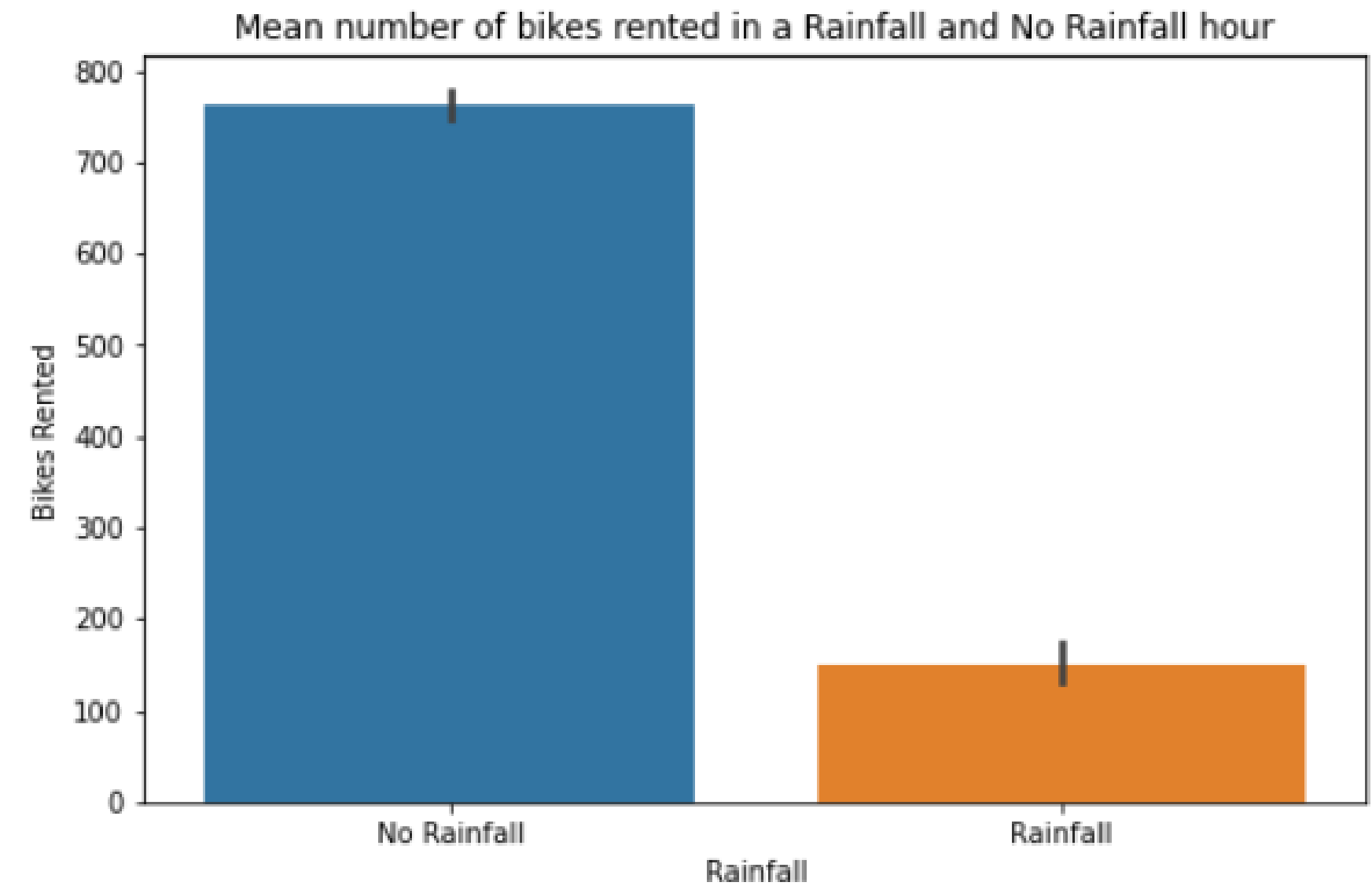


**Number of bikes rented and Snowfall  
does not show a good correlation**

## After Converting Rainfall and Snowfall to categorical variables

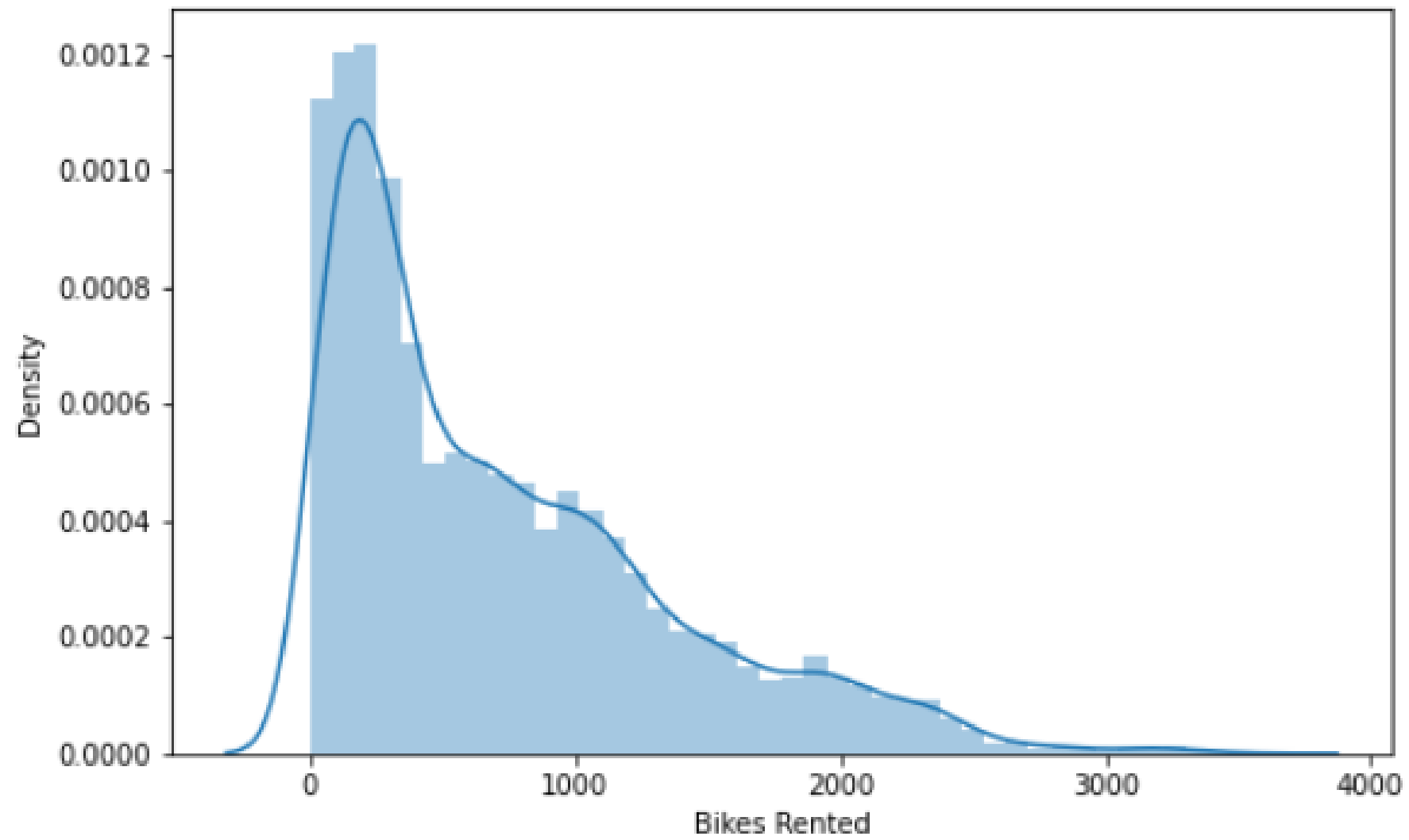


**Maximum number of bikes are rented in No Snowfall hour**

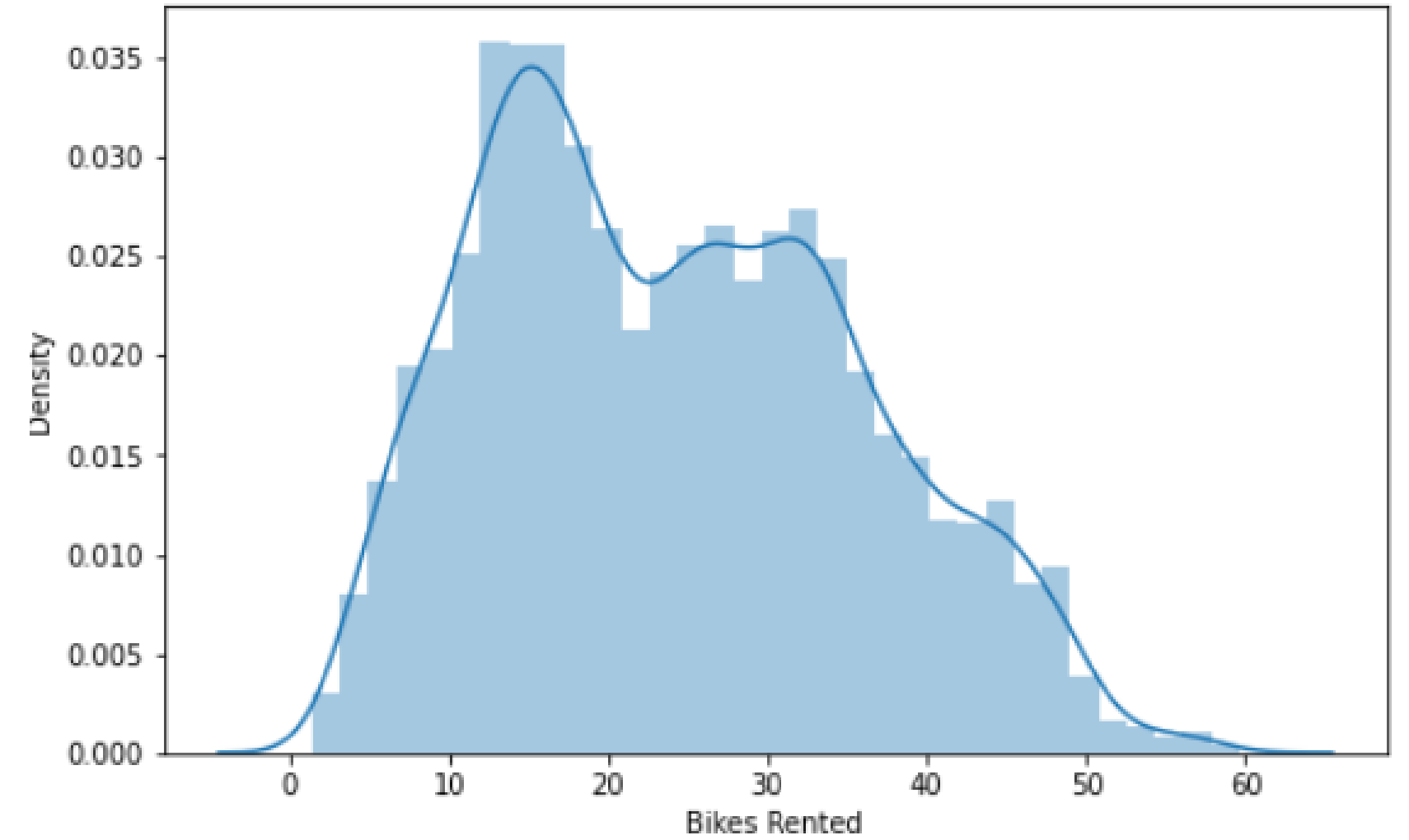


**Maximum number of bikes are rented in No Rainfall hour**

Distribution of the rented bikes



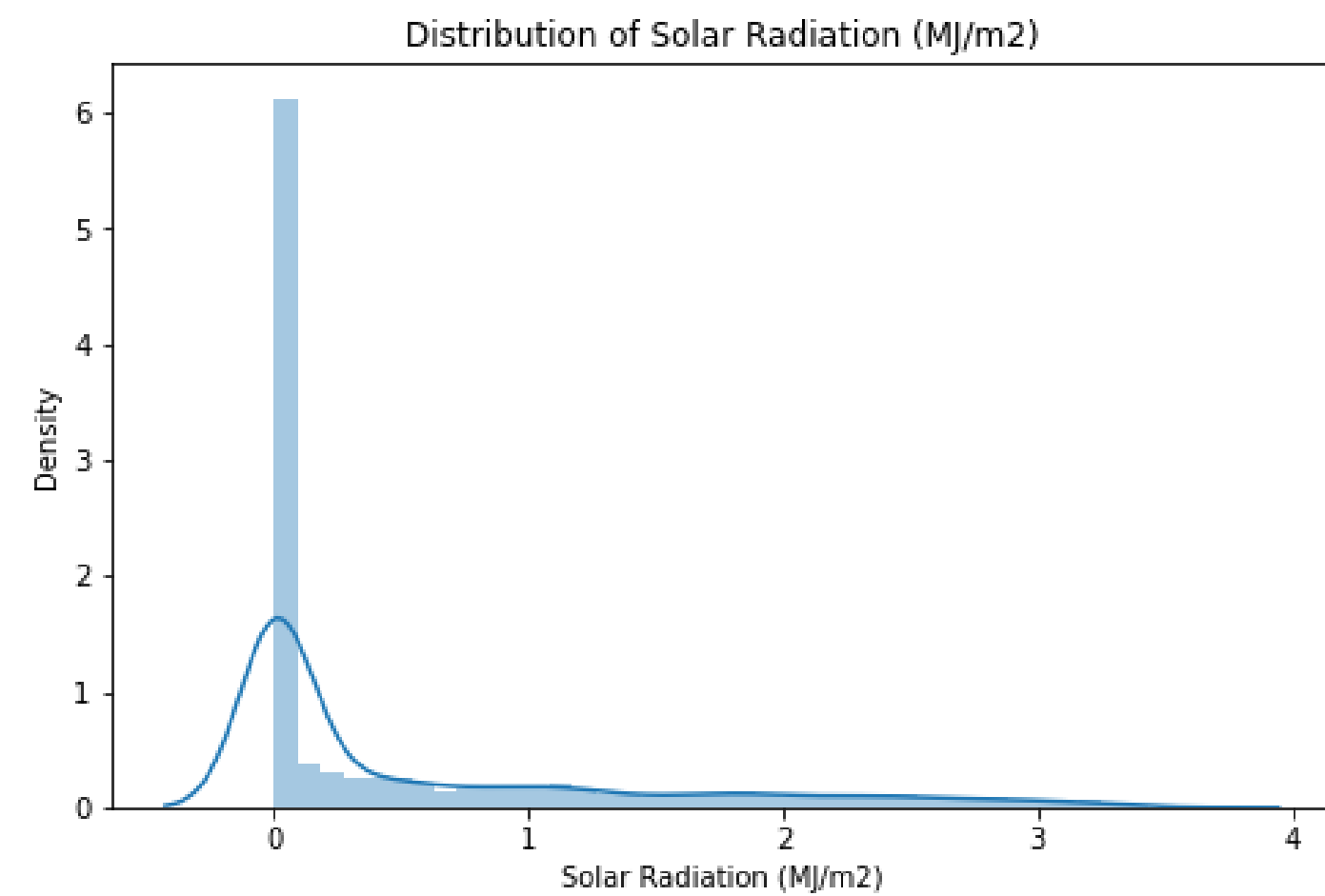
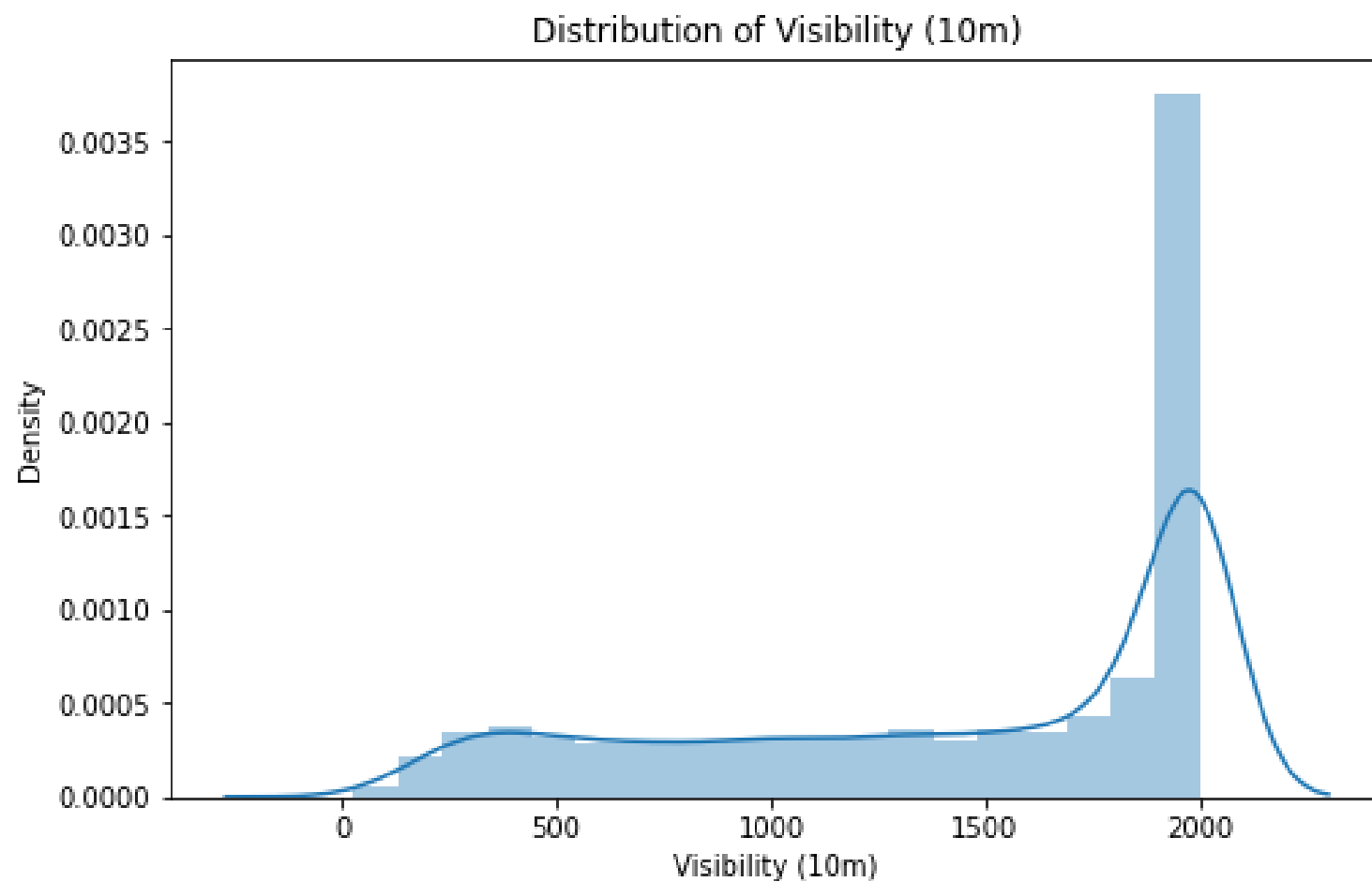
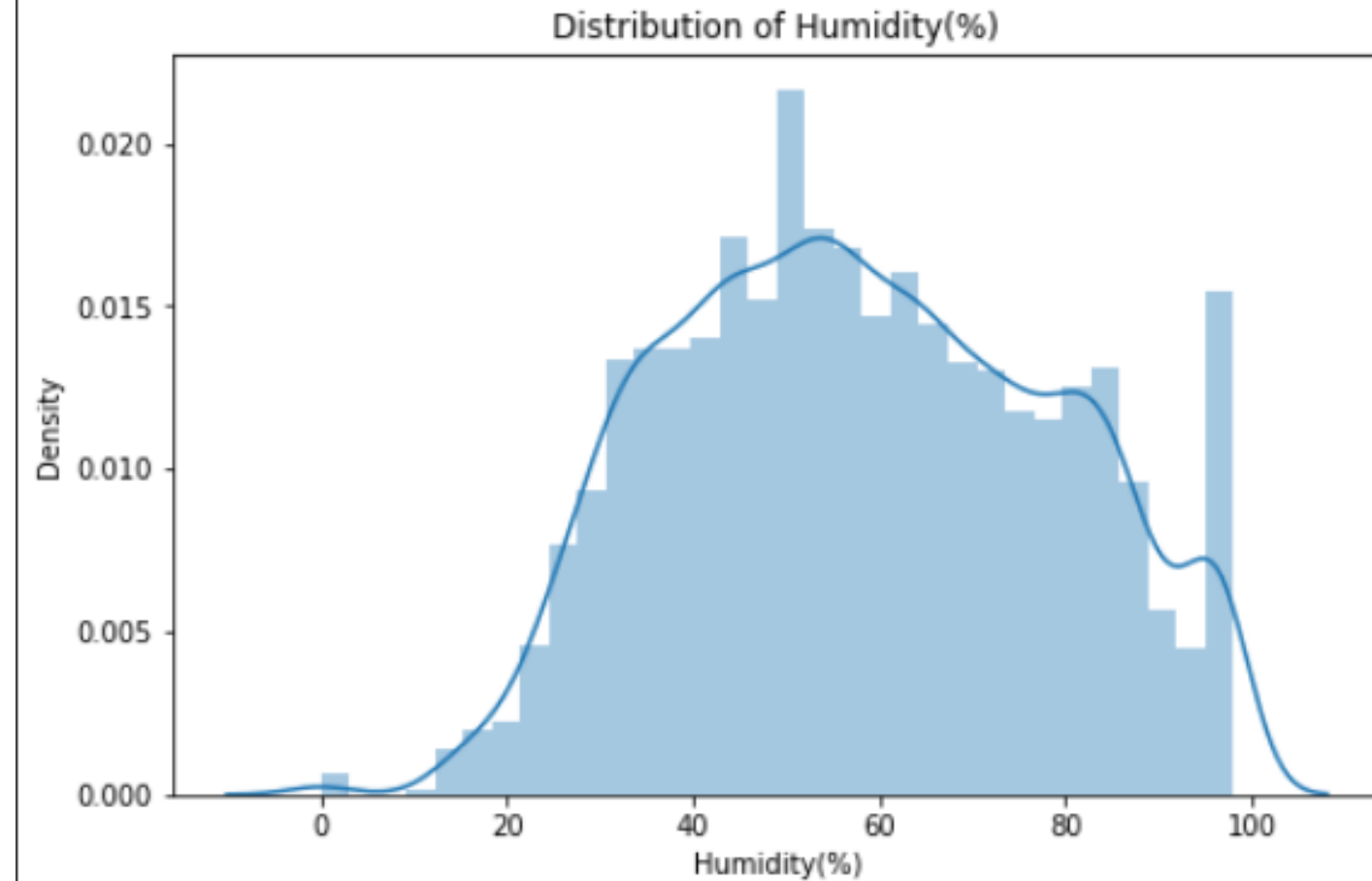
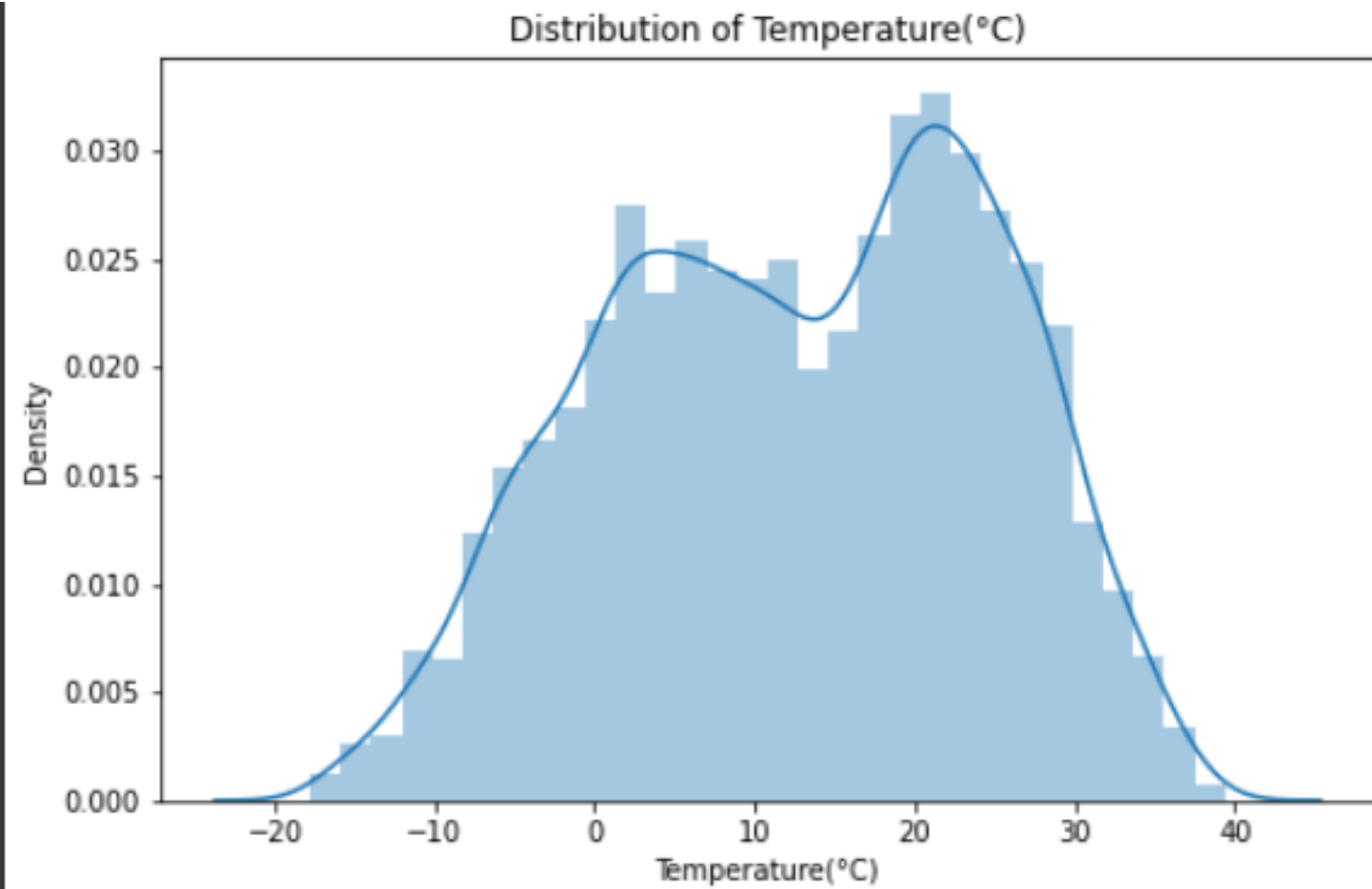
Distribution of square roots of the rented bikes



**It can be seen that the dependent variable (Rented Bike Count) is positively skewed with the skewness value of 1.1397, so transformed it using square-root transformation and made the variable normal with the skewness of 0.346703**



# Distribution of the continuous variables: Temperature, Humidity, Visibility and Solar Radiation



# Data Preparation for Modelling

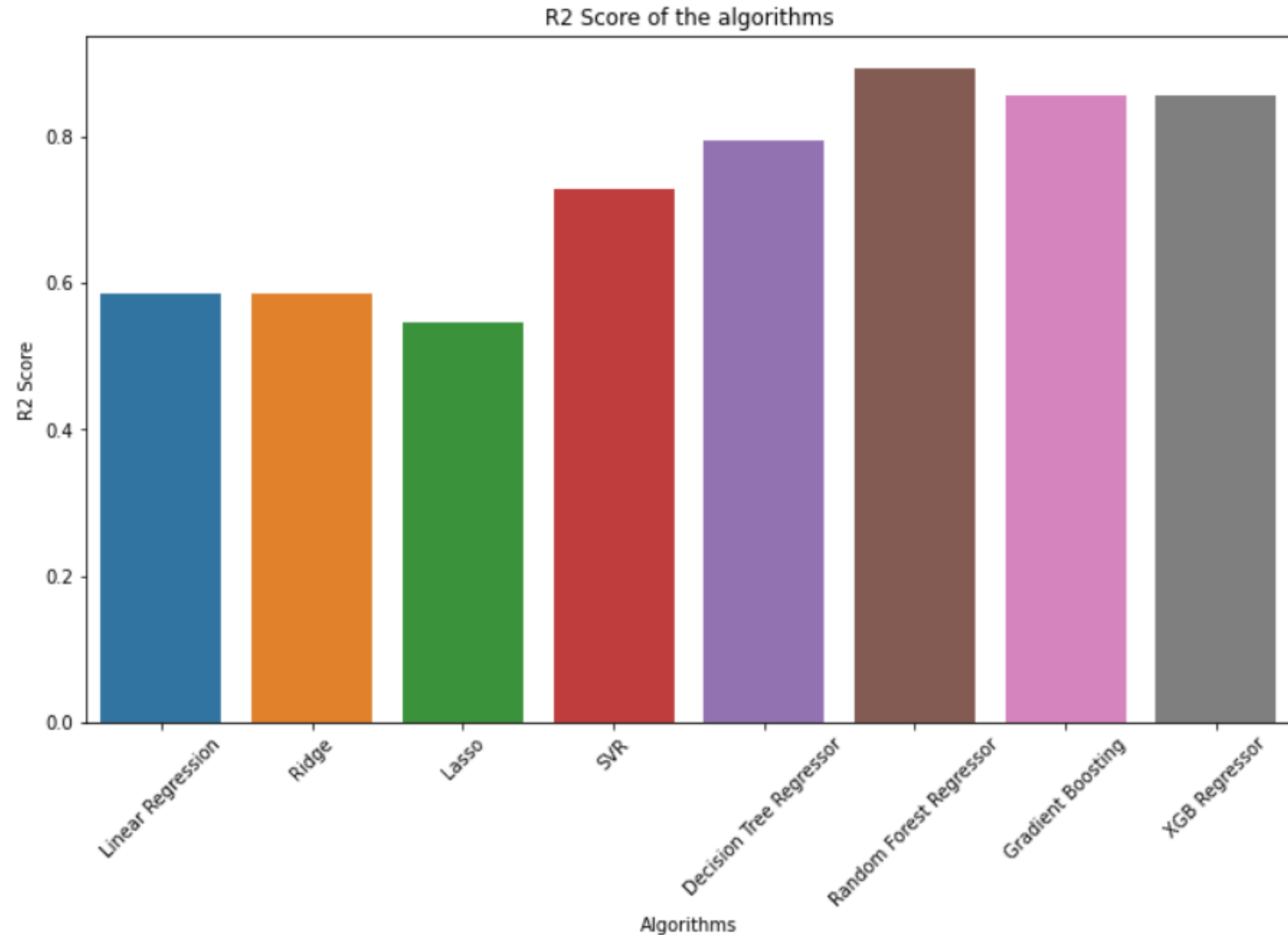
- **Converted 'Month' and 'Day' columns to numerical values.**
- **Converted the columns 'Season', 'Holiday' and 'Year' to numerical values using the 'get\_dummies' function.**
- **Dropped the columns such as:**
  1. **'Date' as the required information from it was already extracted i.e., month number, day number and year.**
  2. **'Dew Point Temperature' as it caused the problem of multi-collinearity with many variables.**
  3. **'Wind Speed' as it did not make a strong relation with the target variable.**
  4. **'Functioning Day' as this column showed obvious results such as no functioning hour showed no bikes being rented and functioning hour showed bikes being rented.**
- **Created X and Y variables as independent and dependent variables respectively.**
- **Split the data into training and testing dataset with the ratio of 80:20 respectively.**
- **Using Standard Scaler scaled the data in a common data range for better and accurate predictions.**
- **Used power transformer to make the training dataset more gaussian-like.**

# R2 Score of Algorithms

Various regression algorithms were used for training the model using the 'cross\_val\_score' with the cross validation of 10 and using 'R2' for scoring purpose.

The R2 Score of the algorithms was found out to be as follows:

1. Linear Regression: 0.64
2. Ridge Regression: 0.64
3. Lasso Regression: 0.60
4. Support Vector Regressor: 0.75
5. Decision Tree Regressor: 0.81
6. Random Forest Regressor: 0.90
7. Gradient Boosting: 0.87
8. XGBoost Regressor: 0.87



# Hyperparameter Tuning

**There is a list of different machine learning models. They all are different in some way or the other, but what makes them different is nothing but input parameters for the model. These input parameters are named as Hyperparameters. These hyperparameters will define the architecture of the model, and the best part about these is that you get a choice to select these for your model. Of course, you must select from a specific list of hyperparameters for a given model as it varies from model to model.**

**Often, we are not aware of optimal values for hyperparameters which would generate the best model output. So, what we tell the model is to explore and select the optimal model architecture automatically.**

**This selection procedure for hyperparameter is known as Hyperparameter Tuning.**

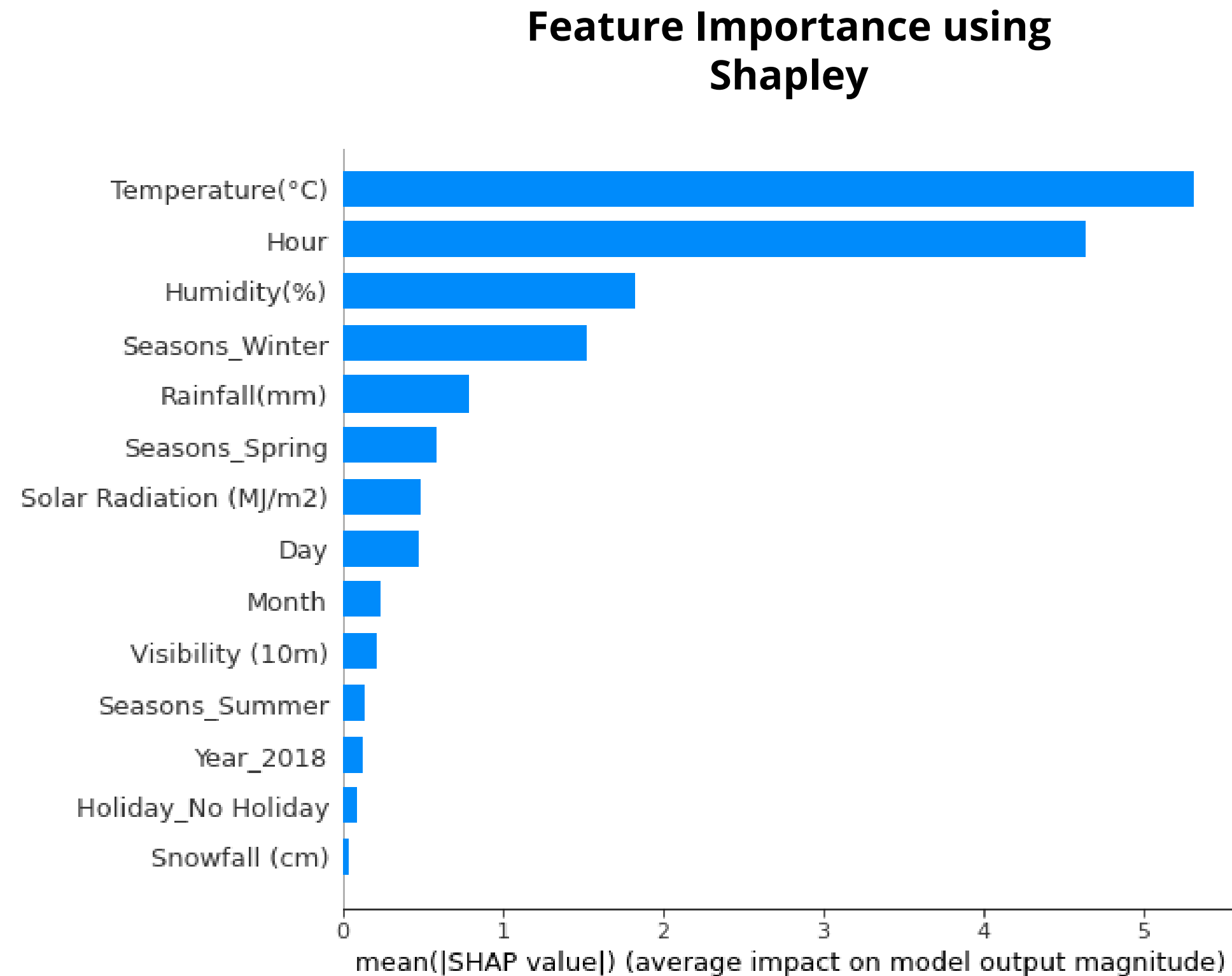


# Hyperparameter tuning of Random Forest Regressor

A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees. Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model.

## Hyperparameters Used:

- **n\_estimators**: Number of trees in the forest
- **max\_depth**: Maximum number of levels in each decision tree
- **min\_samples\_split**: Minimum number of data points placed in a node before the node is split
- **min\_samples\_leaf**: Minimum number of data points allowed in a leaf node
- **bootstrap**: Method for sampling data points (with or without replacement)



**Evaluation Metrics of the Random Forest  
Regressor**

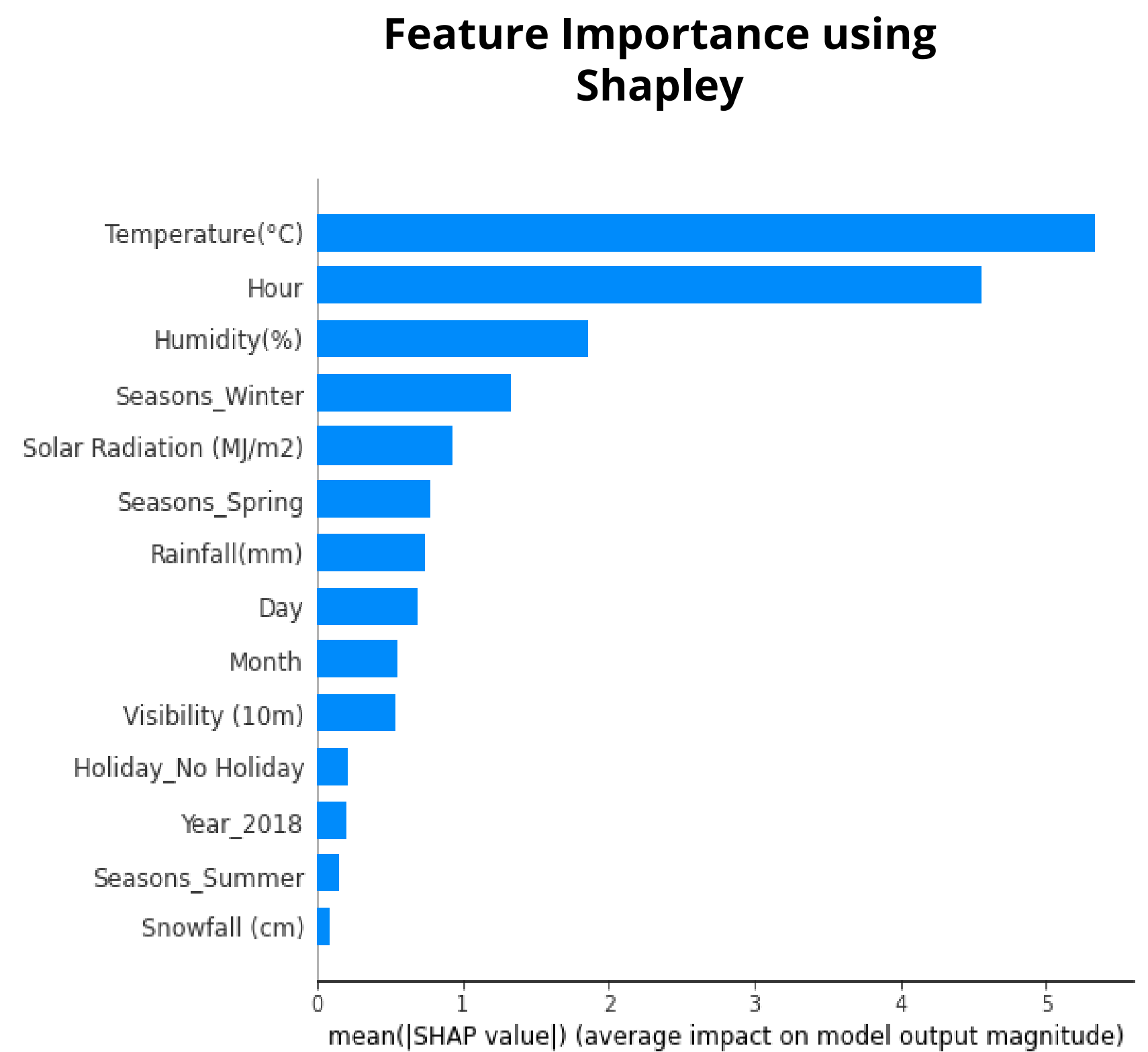
<b>Sr. No.</b>	<b>Evaluation Metrics</b>	<b>Training Score</b>	<b>Testing Score</b>
<b>1.</b>	<b>MSE</b>	<b>5.07</b>	<b>15.16</b>
<b>2.</b>	<b>RMSE</b>	<b>2.25</b>	<b>3.89</b>
<b>3.</b>	<b>R<sup>2</sup></b>	<b>0.96</b>	<b>0.89</b>
<b>4.</b>	<b>MAE</b>	<b>1.45</b>	<b>2.71</b>
<b>5.</b>	<b>MAPE</b>	<b>8.73</b>	<b>15.71</b>
<b>6.</b>	<b>Adjusted R<sup>2</sup></b>	<b>0.96</b>	<b>0.89</b>

# Hyperparameter tuning of Gradient Boosting Regressor

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees. When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees. A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

## Hyperparameters Used:

- **n\_estimators**: Number of trees in the forest
- **max\_depth**: Maximum number of levels in each decision tree
- **min\_samples\_split**: Minimum number of data points placed in a node before the node is split
- **min\_samples\_leaf**: Minimum number of data points allowed in a leaf node
- **max\_features**: The number of features to consider while searching for a best split
- **max\_leaf\_nodes**: The maximum number of terminal nodes or leaves in a tree.
- **learning\_rate**: The learning parameter controls the magnitude of this change in the estimates.



Evaluation Metrics of the Gradient Boosting Regressor

Sr. No.	Evaluation Metrics	Training Score	Testing Score
1.	MSE	0.51	12.49
2.	RMSE	0.71	3.53
3.	$R^2$	0.99	0.91
4.	MAE	0.49	2.44
5.	MAPE	2.76	14.32
6.	Adjusted $R^2$	0.99	0.91

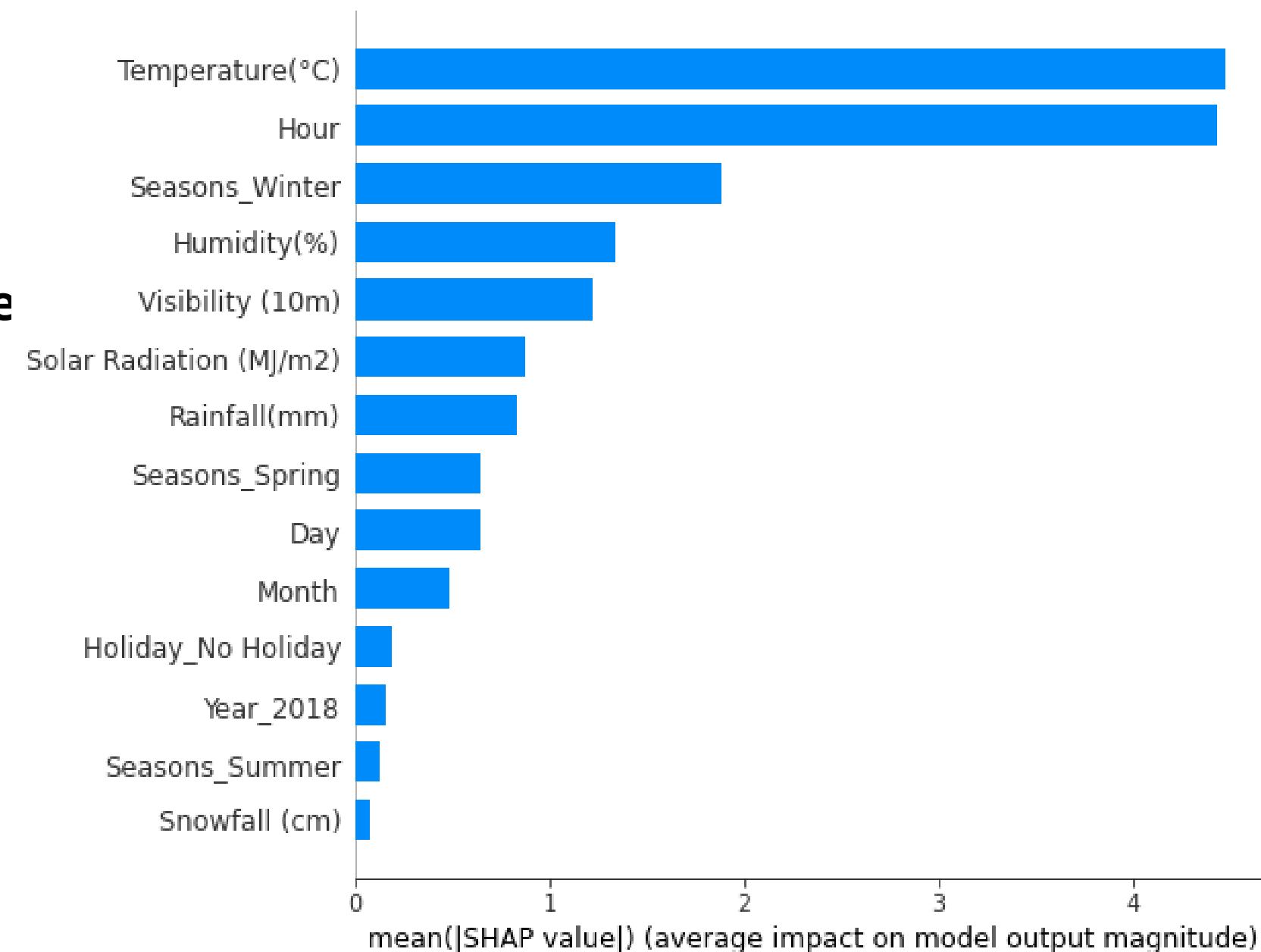
# Hyperparameter tuning of XGBoost Regressor

In this algorithm, decision trees are created in sequential form. Weights play an important role in XGBoost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results. The weight of variables predicted wrong by the tree is increased and these variables are then fed to the second decision tree. These individual predictors then ensemble to give a strong and more precise model. It can work on regression, classification, ranking, and user-defined prediction problems.

## Hyperparameters Used:

- **n\_estimators**: Number of trees in the forest
- **max\_depth**: Maximum number of levels in each decision tree
- **gamma**: It specifies the minimum loss reduction required to make a split.
- **max\_delta\_step**: It allow each tree's weight estimation to be considered
- **min\_child\_weight**: Defines the minimum sum of weights of all observations required in a child.
- **reg\_alpha**: L1 regularization term on weights
- **reg\_lambda**: L2 regularization term on weights
- **learning\_rate**: The learning parameter controls the magnitude of this change in the estimates.

## Feature Importance using Shapley



Evaluation Metrics of the XGBoost Regressor

Sr. No.	Evaluation Metrics	Training Score	Testing Score
1.	MSE	4.33	22.16
2.	RMSE	2.08	4.70
3.	$R^2$	0.96	0.84
4.	MAE	1.49	3.33
5.	MAPE	8.71	18.69
6.	Adjusted $R^2$	0.96	0.84



**Thank  
you**