**A PROJECT REPORT**

**ON**

**BIKE COUNT PREDICTION MODEL**

Submitted in partial fulfillment for the requirement of the award of

TRAINING

IN

Data Analytics, Machine Learning and AI using Python

*Submitted By*

**Soumyabrata Banerjee(**Sikkim Manipal Institue of Technology, Sikkim**)**

*Under the guidance of*

**Bandenawaz Bagwan**

**ACKNOWLEDGEMENT**

My sincere gratitude and thanks towards my project paper guide Bandenawaz Bagwan

It was only with his backing and support that I could complete the report. He provided me all sorts of help and corrected me if ever seemed to make mistakes. I have no such words to express my gratitude.I also thanks Rishab Siromoni sir for his always help and support to make this project successful. He gave me the permission to do the project work. Without his support I couldn’t even start the work. So I am grateful to him. I acknowledge my sincere gratitude to the lecturers, research scholars and the lab technicians for their valuable guidance and helping attitude even in their very busy schedule. And at last but not the least, I acknowledge my dearest parents for being such a nice source of encouragement and moral support that helped me tremendously in this aspect. I also declare to the best of my knowledge and belief that the Project Work has not been submitted anywhere else.

**PREDICTION OF NUMBER OF**

**BIKE-SHARE USERS**

## **A classic regression problem solved with Supervised Machine Learning**

# **Defining the Problem and Project Goal:**

# A bicycle-sharing system is a service in which users can rent/use bicycles available for shared use on a short term basis for a price or free. Currently, there are over 500 bike-sharing programs around the world. Such systems usually **aim to reduce congestion, noise, and air pollution** by providing free/affordable access to bicycles for short-distance trips in an urban area as opposed to motorized vehicles. The number of users on any given day can vary greatly for such systems. The ability to predict the number of hourly users can allow the entities (businesses/governments) that oversee these systems to manage them in a more efficient and cost-effective manner. Our **goal**is to use and optimize Machine Learning models that effectively **predict the number of ride-sharing bikes that will be used in any given 1 hour time-period,** using available information about that time/day.

# See the source image

# **Data-set used:**

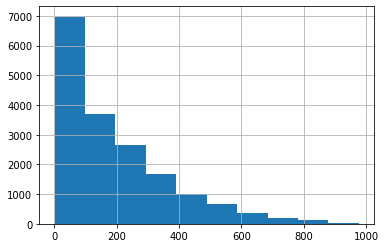
The dataset we are using to build a machine learning model is the bike-sharing dataset that is provided by my training company. It contains both the hourly and daily data about the numbers of bike rentals in Washington, DC between 2011 and 2012. We would use the hourly dataset, which is more complete and have a greater number of observations than the daily dataset.

The dataset has 1 target and 16 features, including both time and weather-related information for each hour on a specific day. All the features and target are listed below:

1. *Record index*
2. *Date*
3. *Season (1:spring, 2:summer, 3:fall, 4:winter)*
4. *Year (0: 2011, 1:2012)*
5. *Month (1 to 12)*
6. Hour (0 to 23)
7. *Holiday : whether that day is holiday or not*
8. *Weekday : day of the week*
9. *Working-day : if day is neither weekend nor holiday , value is 1. Otherwise 0*
10. *Weather situation :  
    — 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 Pallets + Thunderstorm + Mist, Snow + Fog*
11. *Normalized temperature in Celsius. Values are divided to 41 (max)*
12. *Normalized feeling temperature in Celsius. Values are divided to 50 (max)*
13. *Normalized humidity. The values are divided to 100 (max)*
14. *Normalized wind speed. The values are divided to 67 (max)*
15. *Count of casual users*
16. *Count of registered users*
17. *Count of total rental bikes including both casual and registered*

# **Exploratory data analysis:**

Before starting to process a data-set with algorithms, it’s always a good idea to explore it visually. We are going to use **python**for this project. Using the **Matplotlib**and **Seaborne**packages, we can quickly make some plots to investigate how the bicycle usage count is affected by the features available. Now let’s look at some graphs.



The data distribution of the target variable is satisfactory to proceed further. There are sufficient number of rows for each type of values to learn from.

This step is performed to guage the overall data. The volume of data, the types of columns present in the data. Initial assessment of the data should be done to identify which columns are Quantitative, Categorical or Qualitative.

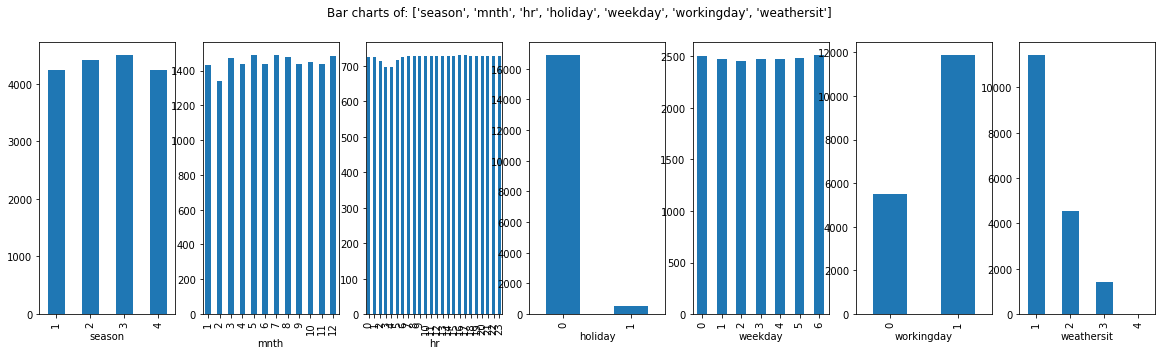
Now based on the basic exploration, we can now create a simple report of the data, noting down our observations regaring each column. Hence, creating a initial roadmap for further analysis.

We can spot a categorical variable in the data by looking at the unique values in them. Typically a categorical variable contains less than 20 Unique values and there is repetition of values, which means the data can be grouped by those unique values.

Based on the Basic Data Exploration, we have spotted seven categorical predictors in the data

**Categorical Predictors:**'season', 'mnth', 'hr', 'holiday', 'weekday', 'workingday', 'weathersit'

We use bar charts to see how the data is distributed for these categorical columns.

****

## Bar Charts Interpretation

These bar charts represent the frequencies of each category in the Y-axis and the category names in the X-axis.

In the ideal bar chart each category has comparable frequency. Hence, there are enough rows for each category in the data for the ML algorithm to learn.

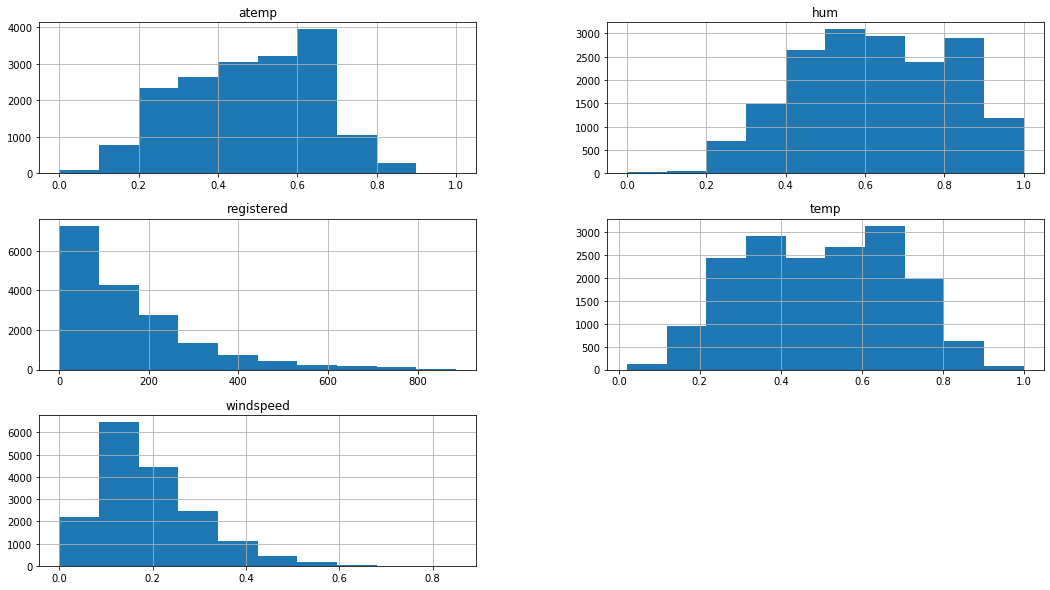
If there is a column which shows too skewed distribution where there is only one dominant bar and the other categories are present in very low numbers. These kind of columns may not be very helpful in machine learning. We confirm this in the correlation analysis section and take a final call to select or reject the column.

In this data, "holiday" is skewed. There is just one bar which is dominating and other categories have very less rows. Such columns may not be correlated with the target variable because there is no information to learn. The algorithms cannot find any rule like when the value is this then the target variable is that. We take a final call for such columns in the correlation section.

**Selected Categorical Variables**: All the categorical variables are selected for further analysis.

'season', 'mnth', 'hr', 'holiday', 'weekday', 'workingday', 'weathersit'

Based on the Basic Data Exploration, There are five continuous predictor variables 'temp','atemp','hum','windspeed', and'registered'.

****

## Histogram Interpretation

Histograms shows us the data distribution for a single continuous variable.

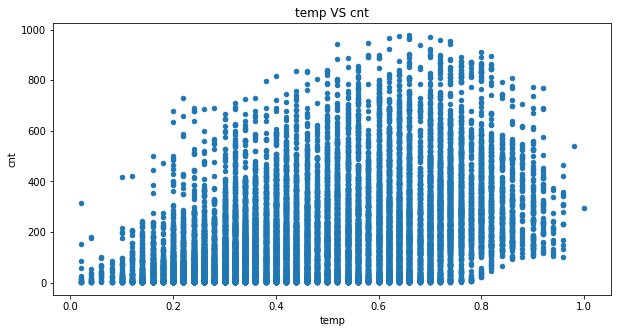
The X-axis shows the range of values and Y-axis represent the number of values in that range. For example, in the above histogram of "atemp", there are around 4000 rows in data that has a value between 0.6 to 0.7.

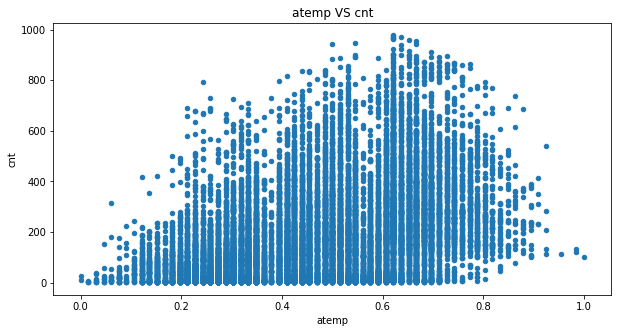
The ideal outcome for histogram is a bell curve or slightly skewed bell curve. If there is too much skewness, then outlier treatment should be done and the column should be re-examined, if that also does not solve the problem then only reject the column.

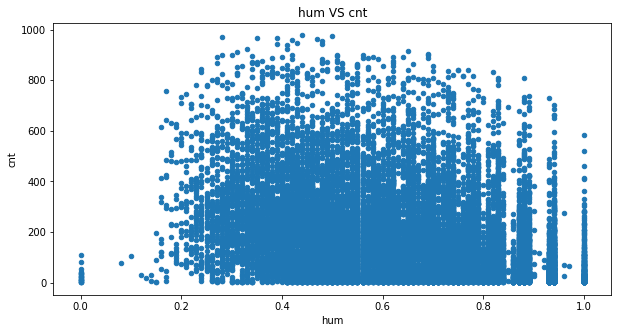
# Feature Selection

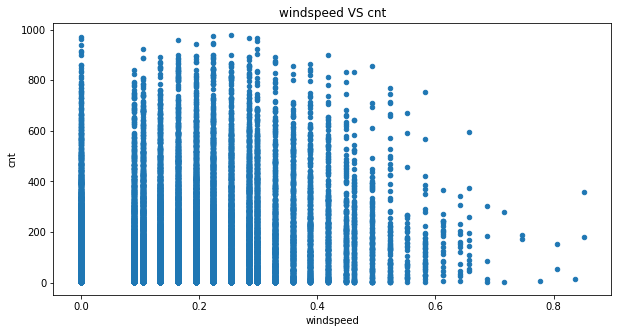
Now its time to finally choose the best columns(Features) which are correlated to the Target variable. This can be done directly by measuring the correlation values. However, it is always helpful to visualize the relation between the Target variable and each of the predictors to get a better sense of data.

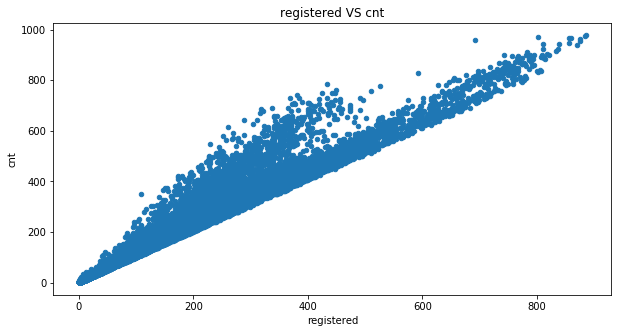
(i). Continuous Vs Continuous -- Scatter Charts



****

****

****

****

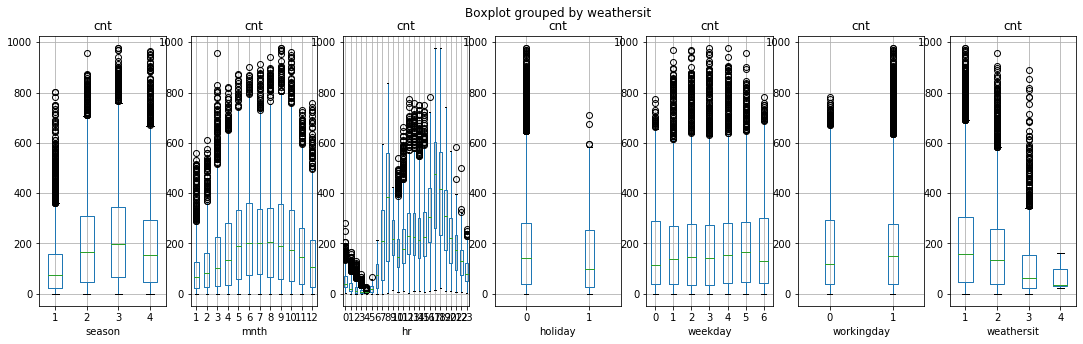
# Scatter charts interpretation

1. Increasing Trend: This means both variables are positively correlated. In simpler terms, they are directly proportional to each other, if one value increases, other also increases. This is good for ML.
2. Decreasing Trend: This means both variables are negatively correlated. In simpler terms, they are inversely proportional to each other, if one value increases, other decreases. This is also good for ML.
3. No Trend: You cannot see any clear increasing or decreasing trend. This means there is no correlation between the variables. Hence the predictor cannot be used for ML.

Based on this chart we can get a good idea about the predictor, if it will be useful or not. We confirm this by looking at the correlation value.

Now, by using the correlation table we can confirm the final feature from the continuous columns and find that registered is having the most correlation value(i.e 0.5>) .

# (ii). Categorical Vs Continuous -- Box Plots



# Box-Plots interpretation

These plots gives an idea about the data distribution of continuous predictor in the Y-axis for each of the category in the X-Axis.

If the distribution looks similar for each category(Boxes are in the same line), that means the the continuous variable has NO effect on the target variable. Hence, the variables are not correlated to each other.

On the other hand if the distribution is different for each category(the boxes are not in same line!). It hints that these variables might be correlated with count.

In this data, all the categorical predictors looks correlated with the Target variable.

So, based on the above analyses of the data we select the final features which we’ll take as the predictors.

**Final Predictors:** Registered, Season, Mnth, Hr, Holiday, Weekday, Workingday, Weathersit.

## **Data Pre-processing for Machine Learning**

List of steps performed on predictor variables before data can be used for machine learning

1. Converting each Ordinal Categorical columns to numeric
2. Converting Binary nominal Categorical columns to numeric using 1/0 mapping
3. Converting all other nominal categorical columns to numeric using pd.get\_dummies()
4. Data Transformation (Optional): Standardization/Normalization/log/sqrt. Important if you are using distance based algorithms like KNN, or Neural Networks.

**Training and Spliting the data for training**

We dont use the full data for creating the model. Some data is randomly selected and kept aside for checking how good the model is. This is known as Testing Data and the remaining data is called Training data on which the model is built. Typically 70% of data is used as Training data and the rest 30% is used as Tesing data.

**Training and Predicting the model using Machine Learning Algorithms:**

For training and finding the prediction of the target variable we have used two types of regressors.

(i). **Multiple Linear Regression:**

Multiple linear regression (MLR), also known simply as multiple regression, is a statistical technique that uses several explanatory variables to predict the outcome of a response variable. The goal of multiple linear regression (MLR) is to model the linear relationship between the explanatory (independent) variables and res

*yi*​=*β*0​+*β*1​*xi*1​+*β*2​*xi*2​+...+*βp*​*xip*​+*ϵ* where, for*i*=*n*observations:*yi*​=dependent variable

*xi*​=explanatory variables*β*0​=y-intercept (constant term) *βp*​ =slope coefficients for each explanatory variable*ϵ*=the model’s error term (also known as the residuals)​

(ii). **Random Forest:**

The random forest algorithm is a supervised classification algorithm. As the name suggests, this algorithm creates the forest with a number of trees.

In general, the **more trees in the forest** the more robust the forest looks like. In the same way in the random forest classifier, the **higher the number** of trees in the forest gives **the high the accuracy** results.

Fitting the above regressors one by one and then observe the accuracy of each regressor.

**Prediction and Accuracy:**

**(i). Multiple Linear Regressor-**

R2 Score: 0.9570683349740267

Mean Accuracy on test data: 32.927670911851834

Mean Accuracy on test data: 86.50306748466258

Final average accuracy of the model: 31.79%

**(ii). Random Forest Regressor-**

R2 Score: 0.9929882007753399

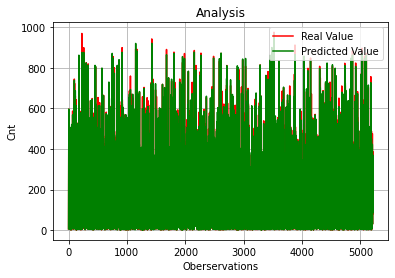
Mean Accuracy on test data: 91.05994081410915

Mean Accuracy on test data: 94.41065127903558

Final average accuracy of the model: 89.33%

**Outcomes:**

After performing the above regressions and training and finding the accuracy of different algorithm we observe Random Forest algorithm shows a better accuracy as compared to linear regression. Hence, we take random forest regressor model as our final regressor.



**Deployment the Model:**

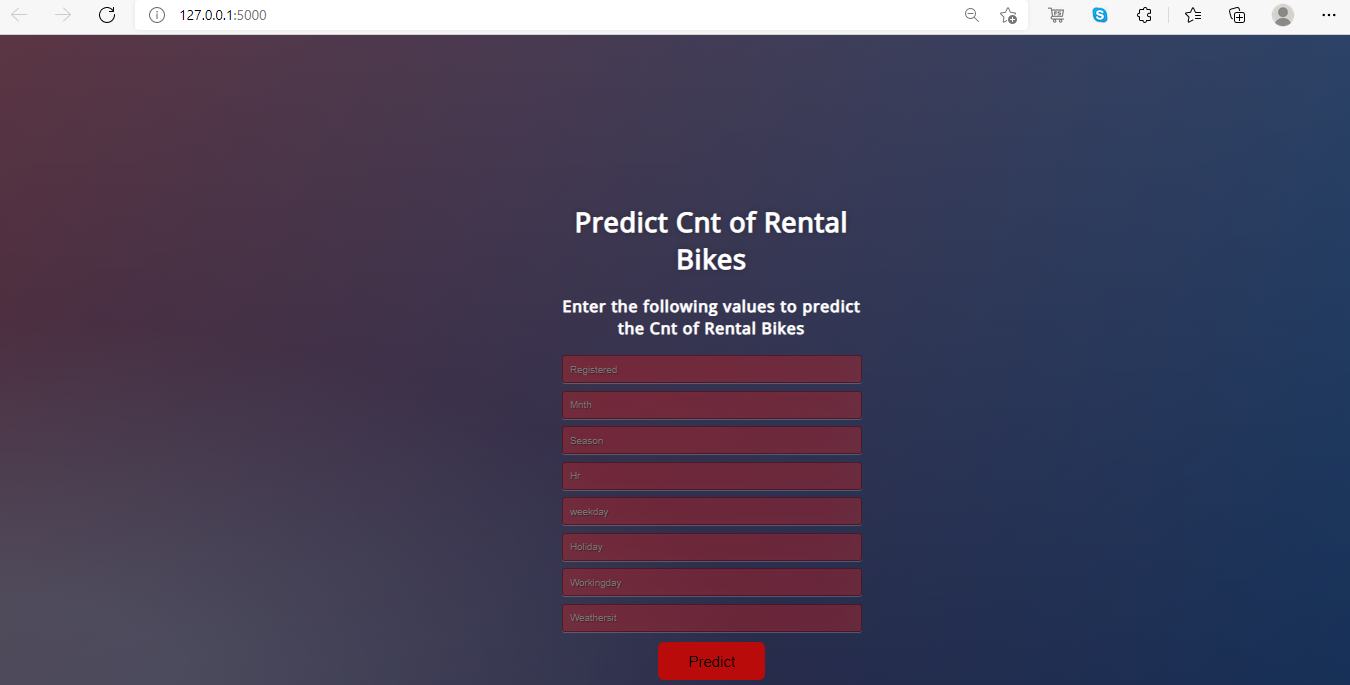
After finalysing the regressor model now we are ready to deploy our model to extend a step towards the production of our ML Model.

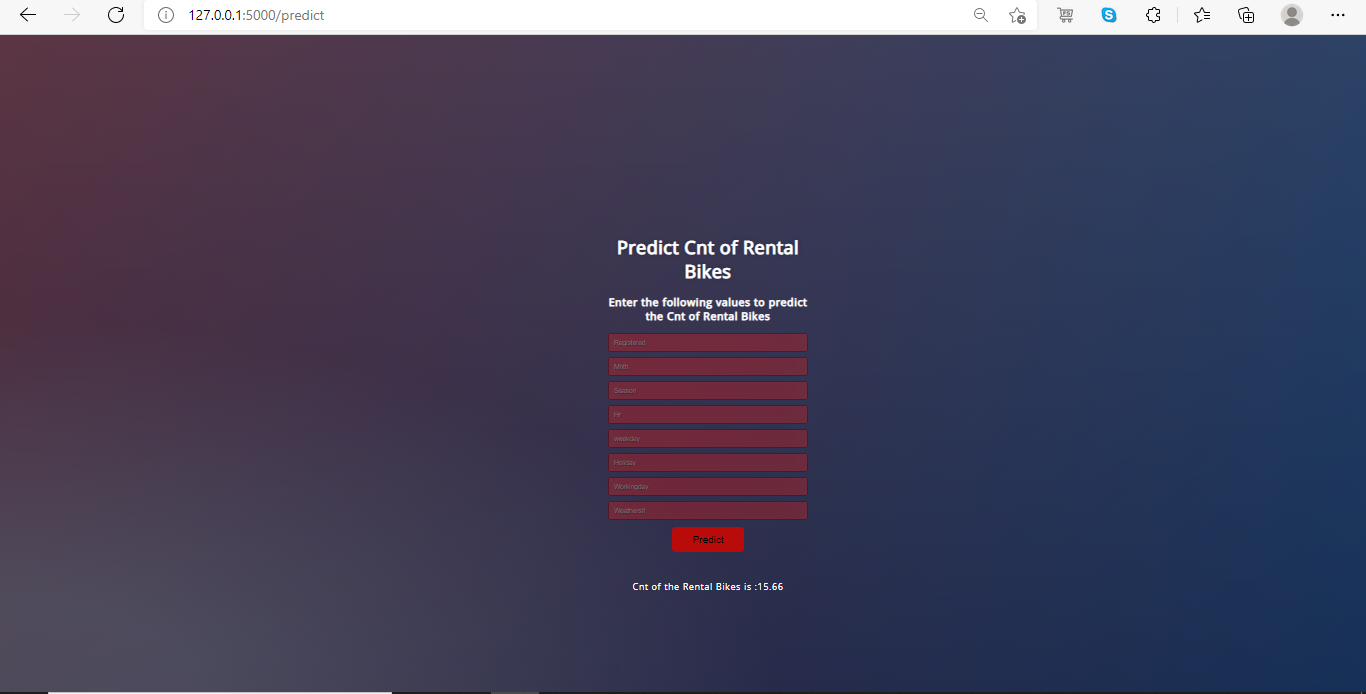
Before deploying the model we need to store the entire model using a pickle file.

After storing the file using pickle then we need to write a suitable code in front end using html and Flask.

Flask is a web framework, it’s a Python module that lets you develop web applications easily. It’s has a small and easy-to-extend core: it’s a microframework that doesn’t include an ORM (Object Relational Manager) or such features.

A web app is created using flask in our local host which produces the target variable(Cnt) on inputs given by the users of their choice.





This local host URL can be deployed for production using the Heroku App which will create an URL which will be open for public domain. It can be accesd by everyone.

**Conclusion:**

Bike sharing systems are new generation of traditional bike rentals where whole process from membership, rental and return back has become automatic. So we successfully build a machine learning model which would the management to deal with the system in more accurate and precise way.

Predictions from Multiple Linear Regression gives very less accuracy which will become inappropriate for the users to deal with. Random Forest Regression prediction results were successfully shows a better accuracy which seems to be more useful for the users.

The overall highest accuracy 89.33% is achieved in predicting the count of the rental bikes process by Random Forest with the sacrifice of significantly extended runtime.

**Bibliography:**

[soumya-678/Bike-Rental-Count-Prediction-Model (github.com)](https://github.com/soumya-678/Bike-Rental-Count-Prediction-Model)