Seoul Bike Sharing Demand Prediction

Sapana Pawar
Data science trainees,
AlmaBetter, Bangalore

Abstract:

This documentation presents a rule-based regression predictive model for bike sharing demand prediction. In recent days, Pubic rental bike sharing is becoming popular because of its increased comfort and environmental sustainability. Data used include Seoul Bike and Capital Bikeshare program data. Both data have weather data associated with it for each hour. For both the dataset, five statistical models were trained with optimized hyperparameters using a repeated cross validation approach and testing set is used for evaluation.

Keywords*:- *Exploratory Data Analysis, Linear Regression, Correlation Analysis, Bike Sharing Demand Prediction.

1.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 the bike count required at each hour for the stable supply of rental bikes.

We have provided the dataset of Seoul Bike Sharing Demand Prediction.

This dataframe presents the rented bike count in this city of Seoul. It is presented as a time series which presents the data with a step of an hour.

For each hour, the dataframe mainly presents weather conditions and information about the day.

Data Description

The dataset contains weather information (Temperature, Humidity, Windspeed, Visibility, Dewpoint, Solar radiation, Snowfall, Rainfall), the number of bikes rented per hour and date information

.Attribute Information:

Following are the features present in the provided datasets.

- Date : The day of the day, during 365 days, type : str
- Rented Bike Count : Number of rented bikes per hour which is the target, type : int
- Hour: The hour of the day, type : int
- Temperature(°C): Temperature per hour, type : Float

- Humidity(%): Humidity in the air in %, type : int
- Wind speed (m/s) : Speed of the wind in m/s, type : Float
- Visibility (10m): Visibility in m, type : int
- Dew point temperature(°C): Temperature at the beginning of the day, type : Float
- Solar Radiation (MJ/m2): Sun contribution, type : Float
- Rainfall(mm): Amount of rain in mm, type : Float
- Snowfall (cm): Amount of snow in cm, type : Float
- Seasons: Season of the year, type : str
- Holiday: If it is holiday period, type: str
- Functioning Day: If it is a Functioning Day, type : str

2. Introduction

Currently rental bikes are introduced in many urban cities for the enhancement of mobility and comfort. The purpose of this movement is to modernize cities and encourage people to head to a green world. Let's take the examples of Paris in 2007, where "velibs" were introduced and Amsterdam, where there are more bikes than cars. The goal is to facilitate the commute in the Seoul and reduce the amount of cars and pollution. Indeed, the development of the way to commute has reduced the use of cars to go to work and visit the city.

It is important to make the rental bike available and accessible to the public, as it provides many alternatives to commuters in metropolises. There are a lot of advantages to bike rents, it is convenient because it permits people not to keep the bike all day long, whether it is at work or at school. Furthermore it is the healthiest way to travel and it has many environmental benefits.

The studied dataset contains *weather* information which are the features (Temperature, Humidity, Wind speed, Visibility, Dew point, Solar radiation, Snowfall, Rainfall), the target is the number of bikes rented per hour and date information. The dataset presents the company's data between December the 1st of 2017 and finishes *one year later*.

How many bikes are rented per hour in function of weather conditions ?

The goal of the company Seoul Bike is providing the city with a stable supply of rental bikes. It becomes a major concern to keep users satisfied. The crucial part is the prediction of bike count rents at each hour for a stable supply of rental bikes. We can suppose that this study could be reported to the company 'Seoul Bikes'. We think it could help them knowing if yes or not they have to supply bike stations in the city, in order to keep the customers.

3. Exploratory Data Analysis(EDA)

- **Importing Python Libraries, Modules and loading dataset**

We have imported some important python libraries and Machine Learning Modules that perform EDA and do prediction for a given dataset.

For analysis,manipulation and imputation purposes, libraries like Numpy and Pandas are imported.

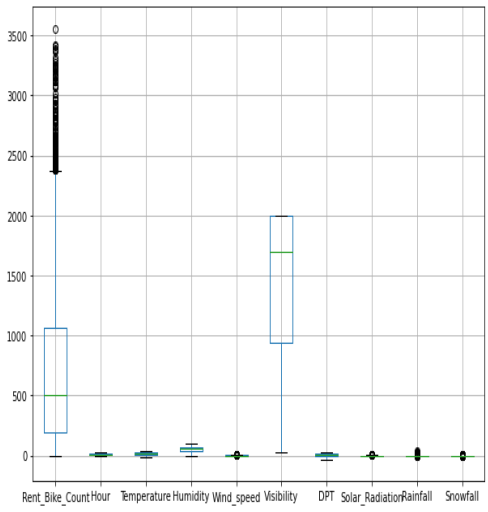
Visual Representation of data makes it easier to identify and share real time trends, outliers and new insights about the information represented in data. For data visualization python libraries like Matplotlib and Seaborn are imported.

- **Exploration**

Data exploration gives some basic information about the dataset. It can include functions like .head and .tail(gives the top and last 5 rows of data), .info, .shape and .describe functions give the basic information,shape and descriptive summary of the dataset.

- **Outliers detection and removal**

After data exploration we looked for outliers present in data by plotting a box plot. An outlier is defined as a data point that is located outside the whiskers of the box plot.



Using IQR Score we have to remove outliers present in the dataset.

IQR

The interquartile range (IQR) is a measure of statistical dispersion and is calculated as the difference between the 75th and 25th percentiles. It is represented by the formula IQR = Q3 − Q1. The lines of code below calculate and print the interquartile range for each of the variables in the dataset.

```
Q1 = dataset.quantile(0.25)
```

```
Q3 = dataset.quantile(0.75)
```

```
IQR = Q3 - Q1
```

The above output prints the IQR scores, which can be used to detect outliers.

IQR Score

This technique uses the IQR scores calculated earlier to remove outliers. The rule of thumb is that anything not in the range of (Q1 - 1.5 IQR) and (Q3 + 1.5 IQR) is an outlier, and can be removed.

```
dataset_clr = dataset[~((dataset < (Q1 - 1.5 * IQR)) |(dataset > (Q3 + 1.5 * IQR))).any(axis=1)]
```

The above line of code below removes outliers based on the IQR range and stores the result in the data frame 'dataset_clr'.

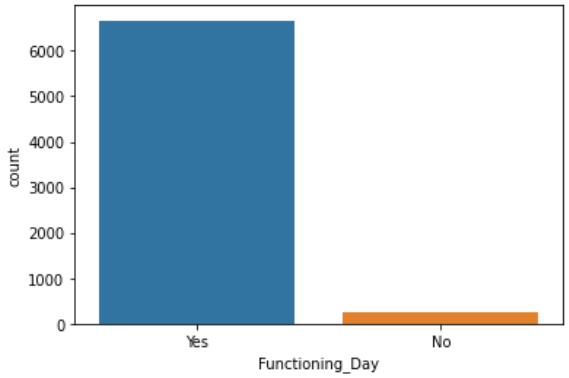
The code dataset_clr.shape prints the shape of this data, which comes out to be 6922 observations of 14 variables. This shows that for our data, a lot of records get deleted.

- **Null Value Treatment**

If our dataset contains a large number of null values which might tend to disturb our accuracy hence we dropped them at the beginning of our project inorder to get a better result.
But here in this dataset there will not be any null value present.

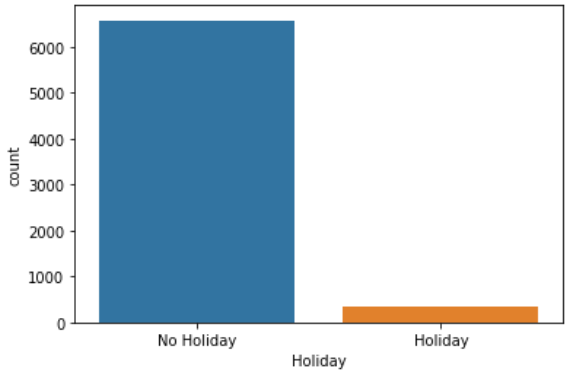
4. Data Visualization

Count plot of Functioning Day



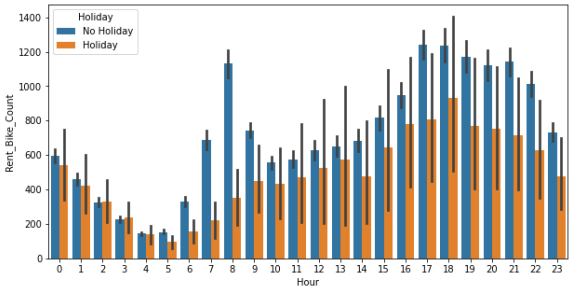
The number of functioning days was more than those days where there were no functioning days.

Count plot of Holiday



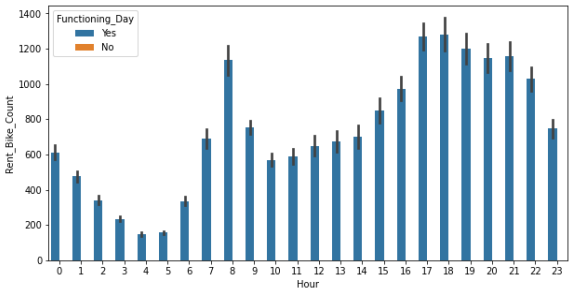
The count of the days when the day was not a holiday is more than the day when the day was a holiday.

Barplot for every hour bike rented on Holiday



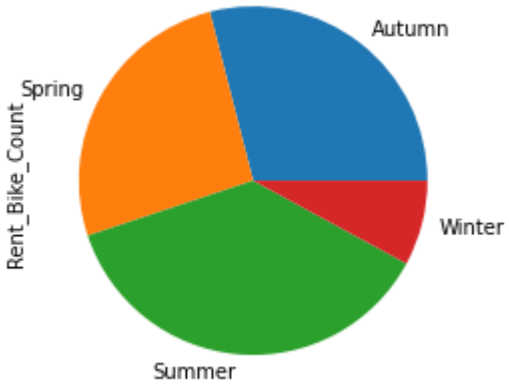
In this plot we can say that when the day was not a holiday, the rental bike count is greater than when the day was a holiday.

Barplot for every hour bike rented on Functioning Day



Here we can see for every hour on the functioning day the bike had rented.But when the day was not the functioning day there was no bike rented.

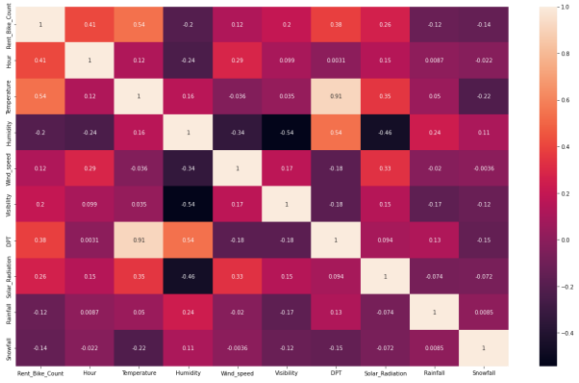
Season Wise Rented bike count



Most of the bikes are rented in the Summer

Season rather than that of Autumn, Spring and Winter.

Correlation Heatmap



As we can see the DPT(Dew Point Temperature) is strongly correlated with Temperature with correlation 0.91.So we have dropped the column 'DPT'.

- **Feature Engineering**

Feature engineering is a machine learning technique that leverages data to create new variables that aren't in the training set. It can produce new features for both supervised and unsupervised learning, with the goal of simplifying and speeding up data transformations while also enhancing model accuracy. Feature engineering is required when working with machine learning models. Regardless of the data or architecture, a terrible feature will have a direct impact on your model.
We have done feature engineering on our dataset to create new features which enhance model accuracy.

- Firstly, we convert the Date column into Datetime Dtype and create the columns Date, Month and Year.
- Nextly, we convert Continuous variables to categorical variables for ease in prediction.

- **Encoding of categorical columns**

1.Label Encoding

Label Encoding is a popular encoding technique for handling categorical variables. In this technique, each label is assigned a unique integer based on alphabetical ordering.
We used the Label Encoding to encode our categorical features because many machine learning algorithms cannot operate on label data directly, They require all input variables and output variables to be numeric.

2. One Hot Encoding

One-Hot Encoding is another popular technique for treating categorical variables. It simply creates additional features based on the number of unique values in the categorical feature. Every unique value in the category will be added as a feature.

One-Hot Encoding is the process of creating dummy variables.

We used One Hot Encoding to produce binary integers of 0 and 1 to encode our categorical features because categorical features that are in string format cannot be understood by the machine and needs to be converted to numerical format.

5. Machine Learning

- **Fitting different models**

For modeling we tried various algorithms like:

1. **Linear Regression**
2. **Lasso Regression**
3. **Ridge Regression**
4. **Elastic Net Regression**
5. **Decision Tree Regression**
6. **XGBoost Regressor**
7. **Random Forest Regressor**

Algorithms:

1. Linear Regression:

Linear regression is a quiet and simple statistical regression method used for predictive analysis and shows the relationship between the continuous variables. Linear regression shows the linear relationship between the independent variable (X-axis) and the dependent variable (Y-axis), consequently called linear regression. The linear regression model gives a sloped straight line describing the relationship within the variables.
To calculate best-fit line linear regression uses a traditional slope-intercept form.

$$y = mx+b \implies y = a_0+a_1x$$

y= Dependent Variable.
x= Independent Variable.
a0= intercept of the line.
a1 = Linear regression coefficient.

2.Lasso Regression:

Lasso stands for Least Absolute Shrinkage Selector Operator. Lasso assigns a penalty to the coefficients in the linear model using the formula below and eliminates variables with coefficients that are zero. This is called shrinkage or the process where data values are shrunk to a central point such as a mean.
Lasso Formula: Lasso = Sum of Error + Sum of the absolute value of coefficients

$$L = \sum(\hat{Y}i - Yi)^2 + \lambda \sum |\beta|$$

Looking at the formula, Lasso adds a penalty equal to the absolute value of the magnitude of the coefficients multiplied by lambda. The value of lambda also plays a key role in how much weight you assign to the penalty for the coefficients. This penalty reduces the value of many coefficients to zero, all of which are eliminated.

2. Ridge Regression:

Ridge regression is a model tuning method that is used to analyze any data that suffers from multicollinearity. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values to be far away from the actual values.
Ridge assigns a penalty that is the squared magnitude of the coefficients to the loss function multiplied by lambda. As Lasso does, ridge also adds a penalty to coefficients the model overemphasizes. The value of lambda also plays a key role in how much weight you assign to the penalty for the coefficients. The larger your value of lambda, the more likely your coefficients get closer and closer to zero. Unlike lasso, the

ridge model will not shrink these coefficients to zero.
Ridge Formula: Sum of Error + Sum of the squares of coefficients

$$L = \sum(\hat{Y}i - Yi)^2 + \lambda \sum \beta^2$$

4. Elastic Net Regression:

Elastic Net combines characteristics of both lasso and ridge. Elastic Net reduces the impact of different features while not eliminating all of the features.
The formula as you can see below is the sum of the lasso and ridge formulas.
Elastic Net Formula: Ridge + Lasso

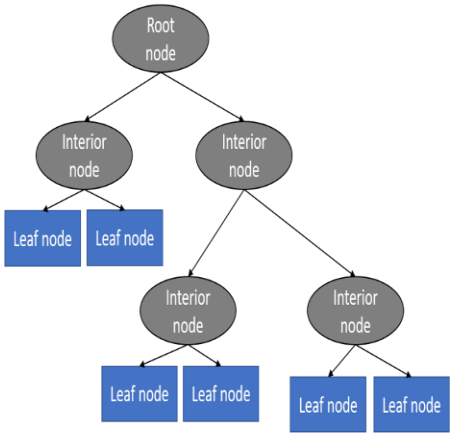
$$L = \sum(\hat{Y}i - Yi)^2 + \lambda \sum \beta^2 + \lambda \sum |\beta|$$

To conclude, Lasso, Ridge, and Elastic Net are excellent methods to improve the performance of your linear model. This includes if you are running a neural network, a collection of linear models. Lasso will eliminate many features, and reduce overfitting in your linear model. Ridge will reduce the impact of features that are not important in predicting your y values. Elastic Net combines feature elimination from Lasso and feature coefficient reduction from the Ridge model to improve your model's predictions.

5. Decision Tree Regressor:

Decision Tree is one of the most commonly used, practical approaches for supervised learning. It can be used to solve both Regression and Classification tasks with the latter being put more into practical application.

It is a tree-structured classifier with three types of nodes. The *Root Node* is the initial node which represents the entire sample and may get split further into further nodes. The *Interior Nodes* represent the features of a data set and the branches represent the decision rules. Finally, the *Leaf Nodes* represent the outcome. This algorithm is very useful for solving decision-related problems.



6. XGBoost:

To understand XGBoost we have to know about gradient boosting beforehand.

- **Gradient Boosting-**

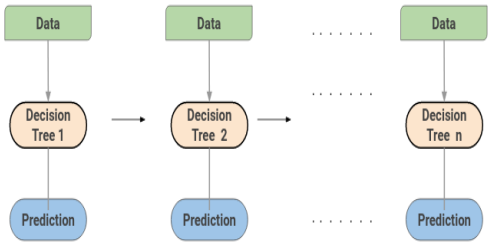
A Gradient Boosting Machine or GBM combines the predictions from multiple decision trees to generate the final predictions. Keep in mind that all the weak learners in a gradient boosting machine are decision trees.

But if we are using the same algorithm, then how is using a hundred decision trees better than using a single decision tree? How do different decision trees capture different signals/information from the data?

Here is the trick – the nodes in every decision tree take a different subset of features for selecting the best split. This

means that the individual trees aren't all the same and hence they are able to capture different signals from the data.

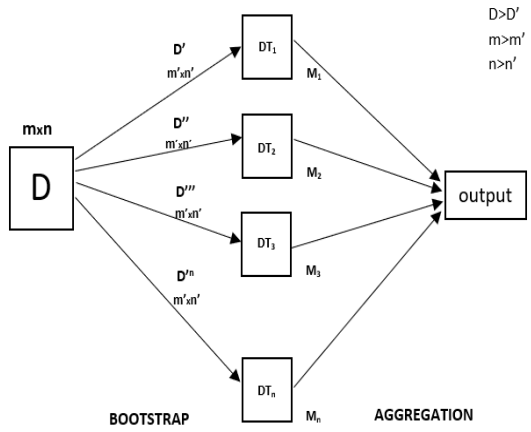
Additionally, each new tree takes into account the errors or mistakes made by the previous trees. So, every successive decision tree is built on the errors of the previous trees. This is how the trees in a gradient boosting machine algorithm are built sequentially.



XGBoost is one of the fastest implementations of gradient boosting. trees. It does this by tackling one of the major inefficiencies of gradient boosted trees: considering the potential loss for all possible splits to create a new branch (especially if you consider the case where there are thousands of features, and therefore thousands of possible splits). XGBoost tackles this inefficiency by looking at the distribution of features across all data points in a leaf and using this information to reduce the search space of possible feature splits.

7.Random Forest:

Every decision tree has high variance, but when we combine all of them together in parallel then the resultant variance is low as each decision tree gets perfectly trained on that particular sample data and hence the output doesn't depend on one decision tree but multiple decision trees. In the case of a classification problem, the final output is taken by using the majority voting classifier. In the case of a regression problem, the final output is the mean of all the outputs. This part is Aggregation.



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 Bootstrap and Aggregation, commonly known as **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. This part is called Bootstrap.

Evaluation Metrics:

Evaluation metrics are a measure of how good a model performs and how well it approximates the relationship. Let us look at MAE, MSE, R-squared and RMSE.

1.Mean Absolute Error (MAE):
This is simply the average of the absolute difference between the target value and the value predicted by the model. Not preferred in cases where outliers are prominent.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$$

MAE does not penalize large errors.

2.Mean Squared Error (MSE):
The most common metric for regression tasks is MSE. It has a convex shape. It is the average of the squared difference between the predicted and actual value. Since it is differentiable and has a convex shape, it is easier to optimize.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

MSE penalizes large errors.

3.Root Mean Squared Error (RMSE):
This is the square root of the average of the squared difference between the predicted and actual value.
R-squared error is better than RMSE. This is because R-squared is a relative measure while RMSE is an absolute measure of fit (highly dependent on the variables — not a normalized measure).

Basically, RMSE is just the root of the

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

RMSE penalizes large errors.

4.R – Squared (R2):
This metric represents the part of the variance of the dependent variable explained by the independent variables of the model. It measures the strength of the relationship between your model and the dependent variable.
The ratio of the residual error (RSS) against the total error (TSS) tells you how much of The total error remains in your regression model. Subtracting that ratio from 1 gives how much error you removed using the regression analysis. That is the R-squared error.
If R^2 is high (say 1), then the model represents the variance of the dependent variable.
If R^2 is very low, then the model does not represent the variance of the dependent variable and regression is no better than taking the mean value, i.e. you are not using any information from the other variables.
A Negative R^2 means you are doing worse than the mean value. It can have a negative value if the predictors do not explain the dependent variables at all such that RSS ~ TSS.

Hyper parameter tuning:

Hyperparameters are sets of information that are used to control the way of learning an algorithm. Their definitions of impact parameters of the models, seen as a way of

learning, change from the new hyperparameters. This set of values affects performance, stability and interpretation of a model. Each algorithm requires a specific hyperparameters grid that can be adjusted according to the business problem. Hyperparameters alter the way a model learns to trigger this training algorithm after parameters to generate outputs.

We used Grid Search CV and Randomized Search CV for hyperparameter tuning. This also results in cross validation and in our case we divided the dataset into different folds. The best performance improvement among the three was by Grid Search CV.

1.Grid Search CV:
Grid Search combines a selection of hyperparameters established by the scientist and runs through all of them to evaluate the model's performance. Its advantage is that it is a simple technique that will go through all the programmed combinations. The biggest disadvantage is that it traverses a specific region of the parameter space and cannot understand which movement or which region of the space is important to optimize the model.

2.Randomized Search CV:
In Random Search, the hyperparameters are chosen at random within a range of values that it can assume. The advantage of this method is that there is a greater chance of finding regions of the cost minimization space with more suitable hyperparameters, since the choice for each iteration is random. The disadvantage of this method is that the combination of hyperparameters is beyond the scientist's control

8. Conclusion:

That's it! We reached the end of our exercise.
Starting with loading the data so far we have done EDA , outlier treatment, feature engineering, encoding of categorical columns, and then model building.

We will see one by one the model with it's R2 score and then we will conclude our conclusion.

Linear Regression: In this model our R2 score is 56.31%, which is not the best score for our predicting model.

Lasso, Ridge and Elastic Net Regression: Lasso and Ridge regression gives the same R2 score as Linear Regression i.e.56.31%.And Elastic Net Regression gives 53.05% .

Decision Tree Regressor: Umm…Here when we implement this model it seems that there is slight increase in the value of R2 score i.e.67.64%, which will not be the best score but that will be satisfactory.

XGBoost: When we implemented the XGboost regressor we got the best R2 score for our model with 83.37%.

Random Forest: Wow...It's outstanding, Here we got the best R2 score value i.e. 86.18%. Which we can say that Random

forest is our best predicting model for regression.
Lastly, to increase the R2 score of this model we have done hyperparameter tuning using GridSearchCV and RandomsearchCV And we got a slight change in the R2 score.
Likewise:
GridSearchCV R2 score is 86.18%.
RandomSearchCV R2 score is 86.26%.

So the R2 score of our best model Random Forest Regressor using RandomSearchCV is 86.26% which can be said to be good for this dataset.

References:

1. Towards Data Science
2. GeeksforGeeks
3. Analytics Vidhya