Bike Sharing Demand Prediction

Anis Bagwan
Ankit Rai
Data science trainees,
AlmaBetter, Bangalore

ABSTRACT:

A bike-sharing systems promotes enhancement of sustainable and comfortable mobility in urban areas. In bike-sharing system, bikes are made available for shared use to individuals which offers a convenient and easy-to-use service for short-distance trips.

To make above benefits available for the larger population, rental bikes are currently 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.

Our job is to perform data analysis and develop a regression model based on the given data, to predict the shared bike demand. This will help the management to strategize their plan and to meet user's Bike Sharing Demand on time.

PROBLEM STATEMENT

In Rented Bike sharing system, the demand keeps fluctuating which means it varies over time depending on various factors. This fluctuating demand may cause the uneven distribution of availability and could possibly lead to the problem of insufficient bikes in certain area for a certain period. For example, bike sharing demand shoot up in the morning and evening for certain period when people commute to their workstation and return to home. Insufficient availability of bike for such time could cause user dissatisfaction followed by the unpopularity of the service.

The main objective is to build a predictive model, which could help the system in predicting shared Bike Demand proactively. This would in turn help the system in matching the user’s demand quickly and efficiently, which ultimately gives user-satisfaction and a successful uninterrupted service.

ABOUT THE DATA

We have been provided past data of Rented Bikes for the analysis. The dataset contains following variables:

- **Date:** Date on which bike was rented
- **Rented Bike count :** It tells number of Bikes rented ,for a given date and for a given one hour duration
- **Hour :** Hour of the day for which data is given
- **Temperature :** Temperature of city in Celsius for the given time duration
- **Humidity - % :** Humidity in the city for the given time duration
- **Windspeed - m/s :** Windspeed in the city for the given time duration
- **Visibility - 10m:** This number tells about the visibility condition of the city
- **Dew point temperature – Celsius :** Dew point temperature of city in Celsius for the given time duration
- **Solar radiation - MJ/m2 :** Tells the Solar Radiation Intensity for a given time
- **Rainfall – mm:** Talks about the Rainfall intensity
- **Snowfall – cm:** Talks about the Snowfall intensity
- **Seasons : Winter, Spring, Summer, Autumn:** Type of Season for which data is taken
- **Holiday - Holiday/No holiday:** Whether the given was a holiday or not
- **Functional Day - NoFunc(Non Functional Hours), Fun(Functional hours) :** For the given day, whether the Bike Sharing System was functioning/working or not

STEPS INVOLVED

- **Feature Engineering:** Feature engineering consists of -

  - **Feature Creation:**

    This method includes identifying the variables that will be most useful in the predictive model.

    We found that “Rented Bike Count” variable is our dependent variable and rest will be independent variables.

    Further, we created three features (Month, Day and Weekend) derived from our existing “Date” variable for the better usability.

  - **Feature Selection**:

    We determined features which were irrelevant or redundant and removed them. “Date” column was irrelevant after we derived important variables from it, and hence we removed it. Further, functioning day ‘column’ was of less importance and hence we dropped it from the dataset.

    We also found redundancy between ‘Temperature’ and ‘Dew Point Temperature’ variable and hence we dropped ‘Dew Point Temperature’ variable.

  - **Transformations:**

    To ensure that the variables are on same scale, we rescaled the numerical variables using MinMaxScaler. The idea was to scale our data into a uniform format that would allow us to utilize the data in a better way while performing fitting and applying.

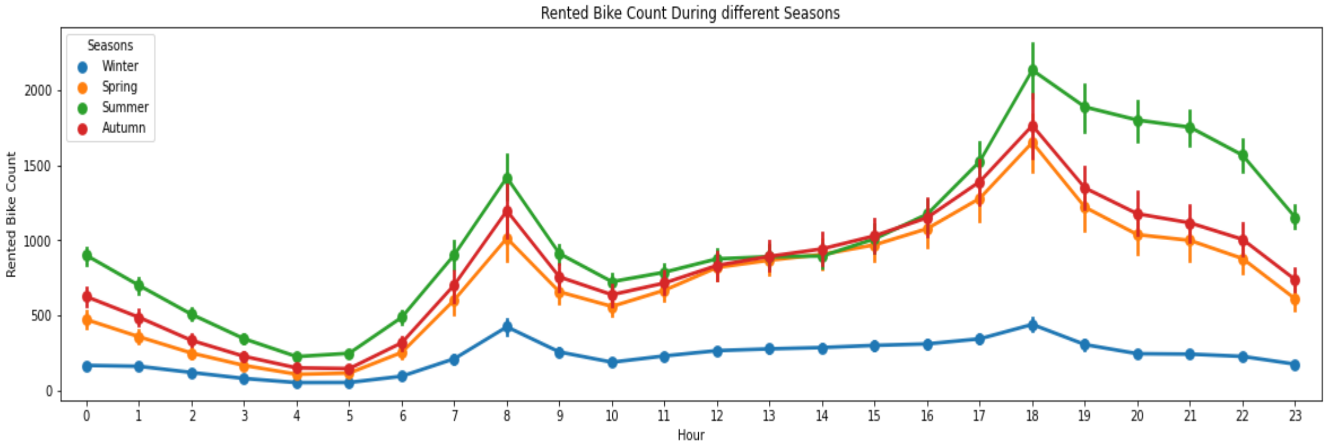
Note:

- We could have done feature extraction, like PCA to reduce the data size. However, since the size of data wasn’t big enough, we skipped it.
- Also, we looked for outliers and found some. But those outliers were not the irrelevant one, and hence we decided to keep them.

- ## Exploratory Data Analysis:

We performed EDA on our data by comparing our dependent variable (“Rented Bike Count”) with other independent variables. This process helped us figuring out various aspects and relationships which impacts on the Bike sharing demand. We also used Bar plots and line graphs for the better representation of the affects of change in variable. Some of the discovered aspects were:

1) The Bike sharing demand shoots up between 5pm to 7pm.
2) The Demand goes maximum in Summer and Minimum in Winter Season
3) The Bike demand is more on non-weekend days.



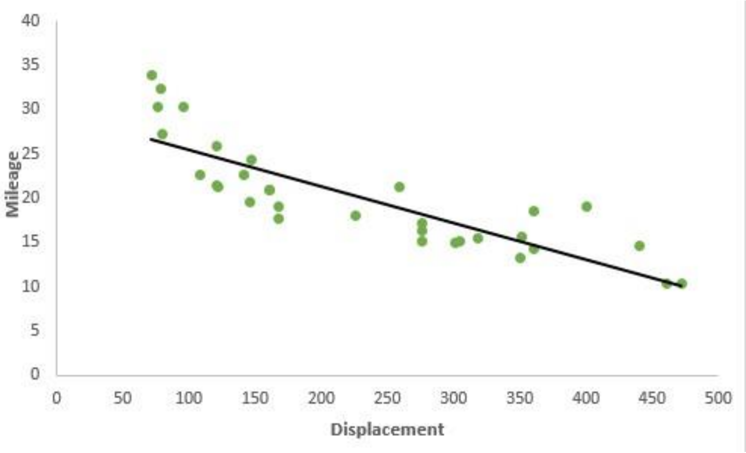
- ## Categorical Feature Encoding

We used One-hot encoding which turns categorical data into a binary vector representation. Pandas “get dummies” makes this very convenient. In this method, for each unique value in a column, a new column is created. The values in this column are represented as 1s and 0s, depending on whether the value matches the column header

- ## Fitting different models

We tried fitting following different models :

1) Multiple Linear Regression:



In this type of regression technique dependent variable and independent variables is assumed to be linear in nature. The equation of multiple linear regression is listed below.

$$y = \beta_1 + \beta_2 X_2 + \beta_3 X_3 + \ldots + \beta_k X_k + \varepsilon$$

2) Elastic Net:

It is a combination of both L1(Lasso) and L2(Ridge) regularization.
The objective function in case of Elastic Net Regression is:

$$\text{Min } (\sum \varepsilon^2 + \lambda_1 \sum \beta^2 + \lambda_2 \sum |\beta|) = \text{Min } \sum (y - (\beta_1 + \beta_2 X_2 + \beta_3 X_3 + \ldots + \beta_k X_k))^2 \lambda_1 \sum \beta^2 + \lambda_2 \sum |\beta|$$

3) Huber Regressor:

Huber regression is a regression technique that is robust to outliers. The idea is to use a different loss function (Huber Loss Function) rather than the traditional least squares. This function is identical to the least square's penalty for small residuals, but on large residuals, its penalty is lower and increases linearly rather than quadratically. It is thus more forgiving of outlier

4) ***Gradient Boosting:***

In gradient boosting, each predictor corrects its predecessor's error.
is a flexible non-parametric statistical learning technique
for classification and regression.

5) ***Random Forest:***

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and average it in case of regression.

- **Model Performance Metrics**

  - **RMSE:**

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

The RMSE tells us how well a regression model can predict the value of the response variable in absolute terms. It measures the standard deviation of residuals.

  - **R2_score:**

$$R^2 = 1 - \frac{\Sigma(y_i - \hat{y})^2}{\Sigma(y_i - \bar{y})^2}$$

R2 tells us how well a model can predict the value of the response variable in percentage terms.

Conclusion

After loading our data, we performed feature engineering on it by removing irrelevant variables and deriving the new ones. Further, we performed EDA and tried drawing insights from the data. Then we hot encoded categorical variables and rescaled the Numerical Variables. After splitting our data into 1:4 ratio for test and split, we applied five regression models. We also used metrics to find the performance of all the applied models. From the metrics result we found that the “Random Forest” model gave the best result , with 97% R2 score.