Bike sharing Prediction

Pavan Potnuru, Mohammed Haseebuddin
Data science trainees,
Alma Better, Bangalore

Abstract:

As a convenient, economical, and ecofriendly travel mode, bike-sharing greatly improved urban mobility. However, it is often very difficult to achieve a balanced utilization of shared bikes due to the asymmetric user demand distribution and the insufficient numbers of shared bikes, docks, or parking areas. If we can predict the short-run bike-sharing demand, it will help operating agencies rebalance bike-sharing systems in a timely and efficient way.

1.Introduction

According to recent studies, it is expected that more than 60% of the population in the world tends to dwell in cities, which is higher than 50% of the present scenario. Some countries around the world are practicing righteous scenarios, rendering mobility at a fair cost and reduced carbon discharge. On the contrary other cities are far behind in the track. Urban mobility usually fills 64% of the entire kilometers travelled in the world. It ought to be modelled and taken over by inter-modality and networked self-driving vehicles which also provides a sustainable means of mobility. Systems called Mobility on Demand have a vital part in raising the vehicles' supply, increasing its idle time and numbers.

2.Problem Statement

We are here to explore the bike sharing dataset and build a model to do the following:

- Maximize: The availability of bikes to the customer.
- Minimize: Minimize the time of waiting to get a bike on rent.

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

3. Feature description

Attribute Information

● Date - year-month-day

● Rented Bike count - Count of bikes rented at each hour

● Hour - Hour of the day

● Temperature- Temperature in Celsius

● Humidity - %

● Wind Speed - m/s

● Visibility - 10m

● Dew point temperature - Celsius

● Solar radiation - MJ/m2

● Rainfall - mm

● Snowfall - cm

● Seasons - Winter, Spring, Summer, Autumn

● Holiday - Holiday/No holiday

● Functional Day – No Func (Non Functional Hours), Fun (Functional hours)

4. Feature Breakdown:

Date: The date of the day, during 365 days from 01/12/2017 to 30/11/2018, formatting in DD/MM/YYYY, we need to convert into date-time format.
Rented Bike Count: Number of rented bikes per hour which our dependent variable and we need to predict that
Hour: The hour of the day, starting from 0-23 it's in a digital time format
Temperature (°C): Temperature of the weather in Celsius and it varies from -17°C to 39.4°C.
Humidity (%): Availability of Humidity in the air during the booking and ranges from 0 to 98%.
Wind speed (m/s): Speed of the wind while booking and ranges from 0 to 7.4m/s.
Visibility (10m): Visibility to the eyes during driving in “m” and ranges from 27m to 2000m.
Dew point temperature (°C): Temperature At the beginning of the day, it ranges from -30.6°C to 27.2°C.
Solar Radiation (MJ/m2): Sun contribution or solar radiation during ride booking which varies from 0 to 3.5 MJ/m2.
Rainfall (mm): The amount of rainfall during bike booking which ranges from 0 to 35mm.
Snowfall (cm): Amount of snowing in cm during the booking 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.

5. Exploratory Data Analysis

Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

The following are the various steps performed as a part of Exploratory Data Analysis:

- Data Preparation
- Data Cleaning
- Univariate Analysis
- Bivariate Analysis
- Multivariate Analysis

5.1 Data Preparation

Firstly, we imported libraries and dataset, some of the libraries used are NumPy, pandas, matplotlib, seaborn, warnings. Once the data is collected, process of analysis begins. But data has to be translated in an appropriate form. This process is known as Data Preparation.

5.2 Data Cleaning

The raw data received in the data set might not be directly suitable for analysis due to presence of unwanted data like, duplicate values, null values, outliers etc. We need to handle them first before we proceed with further analysis.

Removing Duplicates: The dataset provided for this analysis consists of almost 8760 records. None of them are duplicated and all of them are unique. It is better to check for duplicate values before modelling.

Handling null/missing values: It is also possible that the given data set can contain missing information for some or all features in some records, we need to either remove them or find alternatives to fill up the null values. In this data set there are no null values and hence it can be used for modeling.

Feature Handling: We can manipulate some of the features according to our requirements to draw required information from the data. For example, we have divided the date data to extract two new features called, “day_of_the_month” and “month” columns.

5.3 Univariate Analysis: In Univariate Analysis, we choose a single feature from the data and try to determine what the output or the target value is, i.e., one feature/variable at a time.

• Understand the trends and patterns of data
• Analyze the frequency and other such characteristics of data
• Know the distribution of the variables in the data.
• Visualize the relationship that may exist between different variables.

5.4 Bivariate Analysis: In a Bivariate Analysis, we try to analyze two features instead of one, and finally determine the classification of output we are looking for. It is a methodical statistical technique applied to a pair of variables (features/ attributes) of data to determine the empirical relationship between them. In order words, it is meant to determine any concurrent relations.

5.5 Multivariate Analysis: In Multivariate analysis we analyze three or more different features at a time, to understand the relationship between all the features involved.

6. Model Building

6.1 Pre-requisites

Feature Scaling: Scaling data is the process of increasing or decreasing the magnitude according to a fixed ratio, in simpler words you change the size but not the shape of the data. There different three types of feature scaling:

● Centering: The intercept represents the estimate of the target when all predictors are at their mean value, means when x=0, the predictor value will be equal to the intercept.

● Standardization: In this method we centralize the data, then we divide by the standard deviation to enforce that the standard deviation of the variable is one.

$$X_{std} = \frac{X - \bar{X}}{s_X}$$

● Normalization: Normalization most often refers to the process of “normalizing” a variable to be between 0 and 1. Think of this as squishing the variable to be constrained to a specific range. This is also called min-max scaling.

$$X_{norm} = \frac{X - min(X)}{max(X) - min(X)}$$

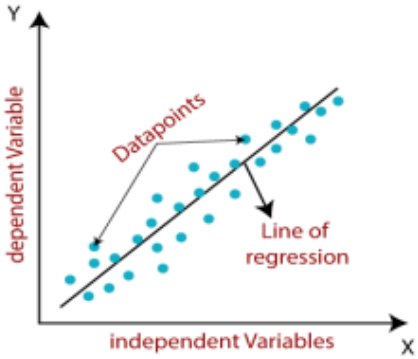
Feature Transformation: Transformation of the skewed variables may also help correct the distribution of the variables. These could be logarithmic, square root, or square transformations. In our dataset Dependent variable i.e., bike_count having a moderate right skewed, to apply linear regression dependent features have to follow the normal

distribution. Therefore, we use square root transformation on top of it.

Detecting Multicollinearity by Variance Inflation Factor (VIF): Variance inflation factor (VIF) is a measure of the amount of multicollinearity in a set of multiple regression variables. Mathematically, the VIF for a regression model variable is equal to the ratio of the overall model variance to the variance of a model that includes only that single independent variable. This ratio is calculated for each independent variable. Where R^2 is the coefficient of determination in linear regression. A higher R-squared value denotes a stronger collinearity. Generally, a VIF above 5 indicates a high multicollinearity. Here we have taken the VIF for consideration value is 5 for having some important features to accord with the model which we will be using in this dataset.

6.2 Model building

Linear Regression: It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, age, product price, etc. Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called linear regression.



Mathematically, we can represent a linear regression as:

Y = b0+B1x+ ε

Y = Dependent Variable (Target Variable)
X = Independent Variable(predictor Variable)
b0 = intercept of the line.
b1 = Linear regression coefficient.
ε = random error

COST FUNCTION(J): Cost function(J) of Linear Regression is the Root Mean Squared Error (RMSE) between predicted y value (pred) and true y value (y).

$$J = \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

Lasso regression: This is a regularization technique used in feature selection using a Shrinkage method also referred to as the penalized regression method. Lasso is short for Least Absolute Shrinkage and Selection Operator, which is used both for regularization and model selection.

$$L_{lasso}(\hat{\beta}) = \sum_{i-1}^{n}(y_i - x_i'\hat{\beta})^2 + \lambda\sum_{j-1}^{m}|\hat{\beta}_j|.$$

If a model uses the L1 regularization technique, then it is called lasso regression.

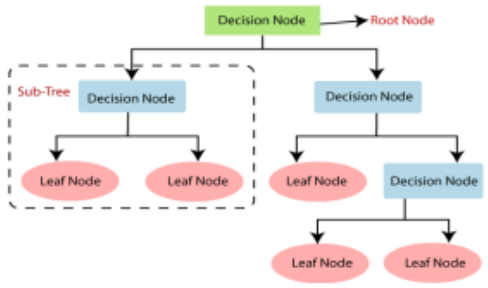
Ridge Regression: Similar to the lasso regression, ridge regression puts a similar constraint on the coefficients by introducing a penalty factor. However, while lasso regression takes the magnitude of the coefficients, ridge regression takes the square.

$$L_{hridge}(\hat{\beta}) = \sum_{i-1}^{n}(y_i - x_i'\hat{\beta})^2 + \lambda\sum_{j-1}^{m}w_j\hat{\beta}_j^2.$$

Ridge regression is also referred to as L2 Regularization.

Decision Tree: Decision Tree is a Supervised learning technique that can be

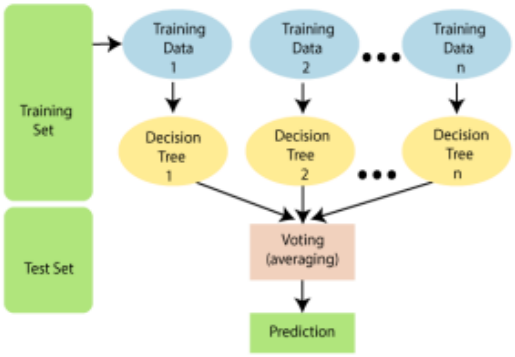
used for both classification and Regression problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm. A decision tree can contain categorical data (YES/NO) as well as numeric data.



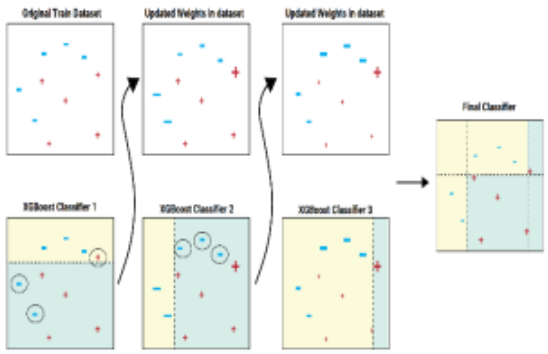
● Root Node: Root node is from where the decision tree starts.
● Leaf Node: Leaf nodes are the final output node
● Splitting: Splitting is the process of dividing the decision node/root node into sub-nodes
● Branch/Sub Tree: A tree formed by splitting the tree.
● Pruning: Pruning is the process of removing the unwanted branches from the tree.
● Parent/Child node: The root node of the tree is called the parent node, and other nodes are called the child nodes.

Ensemble uses two types of methods:
● Bagging– It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example, Random Forest.
● Boosting– It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. For example, ADA BOOST, XG BOOST

Random Forest: Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.



XGBoost Algorithm: 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 classifiers/predictors then ensemble to give a strong and more precise model. XGBoost comes under the boosting ensemble techniques which combines the weakness of primary learners to the next strong and compatible learners.



6.3 Hyperparameter Tuning

Hyper parameters are the variables that the user specify usually while building the Machine Learning model.

Grid Search CV() uses a different combination of all the specified hyperparameters and their values and calculates the performance for each combination and selects the best value for the hyperparameters. This makes the processing time-consuming and expensive based on the number of hyperparameters involved. GridSearchCV() method is available in the scikit-learn class model_selection. It can be initiated by creating an object of GridSearchCV(). Primarily, it takes 4 arguments i.e. estimator, param_grid, cv, and scoring.

- N_ESTIMATORS : The n_estimator parameter controls the number of trees inside the classifier. We may think that using many trees to fit a model will help us to get a more generalized result. The default number of estimators is 100 in scikitlearn.
- MAX_DEPTH: It governs the maximum height up to which the trees inside the forest can grow. It is one of the most important hyperparameters when it comes to increasing the accuracy of the model. The default is set to None.
- MIN_SAMPLES_SPLIT: It specifies the minimum amount of samples an internal node must hold in order to split into further nodes. However, the default value is set to 2.
- MIN_SAMPLES_LEAF: It specifies the minimum amount of samples that a node must hold after getting split. The default value is set to 1.
- ETA/ LEARNING RATE: Learning rate is a hyper-parameter that controls how much we are adjusting the weights of our network with respect the loss gradient.

6.4 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, Adjusted Rsquared, and RMSE.

Mean Absolute Error (MAE)

This is simply the average of the absolute difference between the target value and the value predicted by the model.

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

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.

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

Root Mean Squared Error (RMSE)

This is the square root of the average of the squared difference of the predicted and actual value.

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

R-Squared

R-square is a comparison of residual sum of squares (SSres) with total sum of squares (SStot).

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

Adjusted R-Squared

The main difference between adjusted Rsquared and R-square is that Rsquared describes the amount of variance of the dependent variable represented by every single independent variable, while adjusted R-squared measures variation explained by only the independent variables that actually affect the dependent variable.

$$R^2_{adjusted} = \left[\frac{(1-R^2)(n-1)}{n-k-1}\right]$$

7. Conclusion

Starting with loading the data so far, we have done EDA, outlier treatment, encoding of categorical columns, feature selection and then model building. After trying different models, finally XGBOOST regressor gives us the highest accuracy ranging between 82-96%. Functioning day is the most important feature and Winter is the second most for XGBoostRegressor.