SEOUL RENTAL BIKE DEMAND PREDICTION

Data science trainee,
AlmaBetter, Bangalore

Abstract:

As more number of rented bikes are being used in the cities nowadays, it becomes important for the company to predict the number of required rental bikes required across a day so that no demand supply gap would be generated for rental bikes. This project aims at providing necessary solution to predict the rental bikes demand using machine learning algorithms so that all the stakeholders of the business can be satisfied.

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.

Variables

- 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 - NoFunc(Non Functional Hours), Fun(Functional hours)

Introduction

The present scenario is about how good is the customer service in any industry as the number of options at the customer's disposal is unlimited. So, it becomes extremely important to make sure that the customers will not be made to wait for the rental bikes. It would also not be practical to keep lot of bikes even when the demand is low. Hence, with the help of machine learning, this project aims at predicting the rental bike demand so that no problems arise.

Steps involved:

- **Exploratory Data Analysis**:
  The first step of our project is performing the EDA process on the dataset so that we can get the idea about the dataset i.e. the number of variables, the data type of the variables , visualize the dataset for Better understanding and decide the suitable methods and algorithms that might produce desired outcomes

- **Data Preprocessing:**
  In EDA process we find the type of dataset and decide the approach, in this project the preprocessing steps would removing the punctuations, stopwords , generate count vectorizer and document term matrix which would help in building up the model.

- **Building Machine Learning Model**:
  After the data preprocessing is done then the data will be ready to be fit into machine learning models .For current problem statement topic modeling approach would be suitable . In topic modeling, a topic is defined by a cluster of words with each word in the cluster having a probability of occurrence for the given topic, and different topics have their respective clusters of words along with corresponding probabilities.

Summary:
At last the summary of the project is described to have brief look over the project.

Algorithms:

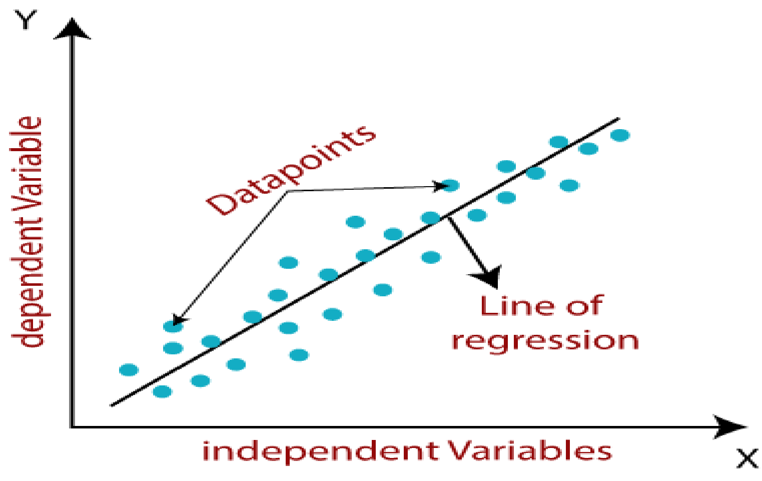
Linear Regression:

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as **sales, salary, age, product price,** etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

The linear regression model provides a sloped straight line representing the relationship between the variables. Consider the below image:

-



$y= a_0+a_1x+ \varepsilon$

Here,

Y= Dependent Variable (Target Variable)
X= Independent Variable (predictor Variable)
a0= intercept of the line (Gives an additional degree of freedom)
a1 = Linear regression coefficient (scale factor to each input value).
ε = random error

The values for x and y variables are training datasets for Linear Regression model representation.

Ridge Regression (L2 Regularization)

This technique performs L2 regularization. The main algorithm behind this is to modify the RSS by adding the penalty which is equivalent to the square of the magnitude of coefficients. However, it is considered to be a technique used when the info suffers from multicollinearity (independent variables are highly correlated). In multicollinearity, albeit the smallest amount squares estimates (OLS) are unbiased, their variances are large which deviates the observed value faraway from truth value. By adding a degree of bias to the regression estimates, ridge regression reduces the quality errors. It tends to solve the multicollinearity problem through shrinkage parameter λ. Now, let us have a look at the equation below.

$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_2^2}_{\text{Penalty}}$$

In this equation, we have two components. The foremost one denotes the least square term and later one is lambda of the summation of β2 (beta- square) where β is the coefficient. This is added to least square term so as to shrink the parameter to possess a really low variance.

Every technique has some pros and cons, so as Ridge regression. It decreases the complexity of a model but does not reduce the number of variables since it never leads to a coefficient tending to zero rather only minimizes it. Hence, this model is not a good fit for feature reduction.

Lasso Regression (L1 Regularization)

This regularization technique performs L1 regularization. Unlike Ridge Regression, it modifies the RSS by adding the penalty (shrinkage quantity) equivalent to the sum of the absolute value of coefficients.

Looking at the equation below, we can observe that similar to Ridge Regression, Lasso (Least Absolute Shrinkage and Selection Operator) also penalizes the absolute size of the regression coefficients. In addition to this, it is quite capable of reducing the variability and improving the accuracy of linear regression models.

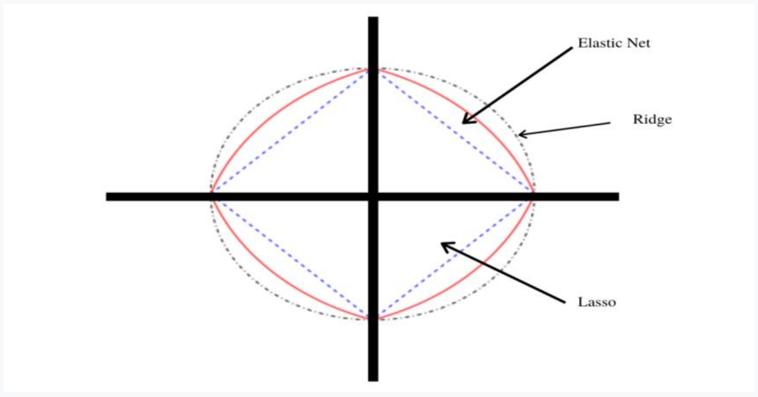
$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_1}_{\text{Penalty}}$$

Limitation of Lasso Regression:

- If the number of predictors (p) is greater than the number of observations (n), Lasso will pick at most n predictors as non-zero, even if all predictors are relevant (or may be used in the test set). In such cases, Lasso sometimes really has to struggle with such types of data.
- If there are two or more highly collinear variables, then LASSO regression select one of them randomly which is not good for the interpretation of data.
- Lasso regression differs from ridge regression in a way that it uses absolute values within the penalty function, rather than that of squares. This leads to penalizing (or equivalently constraining the sum of the absolute values of the estimates) values which causes some of the parameter estimates to turn out exactly zero. The more penalty is applied, the more the estimates get shrunk towards absolute zero. This helps to variable selection out of given range of n variables.

Elastic Net Regression

Elastic net linear regression uses the penalties from both the lasso and ridge techniques to regularize regression models. The technique combines both the lasso and ridge regression methods by learning from their shortcomings to improve the regularization of statistical models.



The elastic net method improves lasso's limitations, i.e., where lasso takes a few samples for high dimensional data. The elastic net procedure provides the inclusion of "n" number of variables until saturation. If the variables are highly correlated groups, lasso tends to choose one variable from such groups and ignore the rest entirely.

To eliminate the limitations found in lasso, the elastic net includes a quadratic expression ($||\beta||^2$) in the penalty, which, when used in isolation, becomes ridge regression. The quadratic expression in the penalty elevates the loss function toward being convex. The elastic net draws on the best of both worlds – i.e., lasso and ridge regression.

In the procedure for finding the elastic net method's estimator, two stages involve both the lasso and regression techniques. It first finds the ridge regression coefficients and then conducts the second step by using a lasso sort of shrinkage of the coefficients.

This method, therefore, subjects the coefficients to two types of shrinkages. The double shrinkage from the naïve version of the elastic net causes low efficiency in predictability and high bias. To correct for such effects, the coefficients are rescaled by multiplying them by ($1+\lambda_2$).

Summary

- **The elastic net method performs variable selection and regularization simultaneously.**
- **The elastic net technique is most appropriate where the dimensional data is greater than the number of samples used.**
- **Groupings and variables selection are the key roles of the elastic net technique.**

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. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

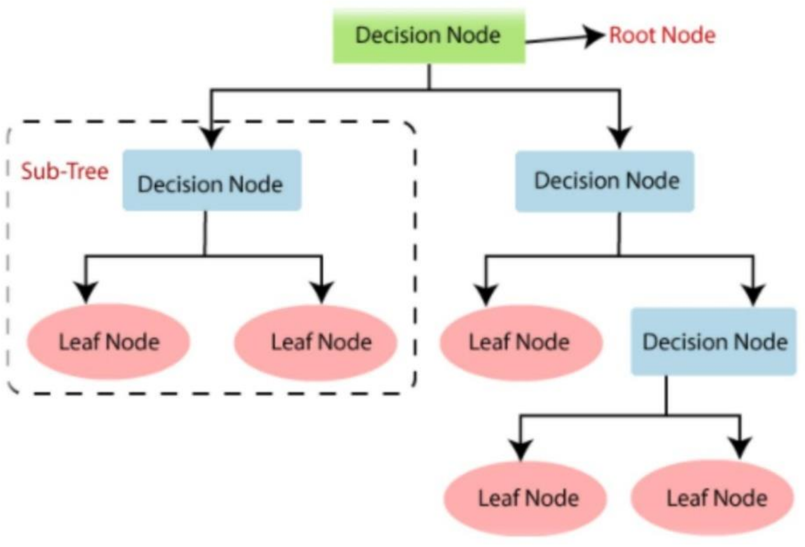


Fig 1.General structure of decision tree

Random Forest:

- Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning**,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.
- 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 random forest leads to higher accuracy and prevents the problem of overfitting.

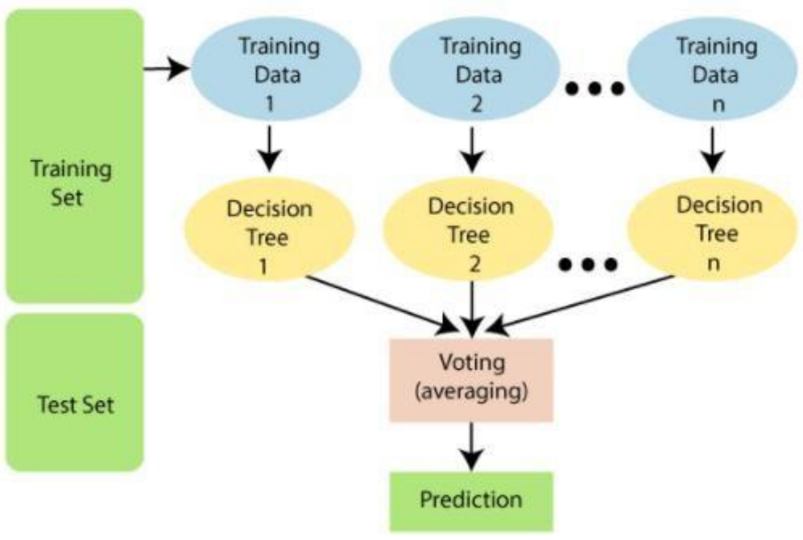
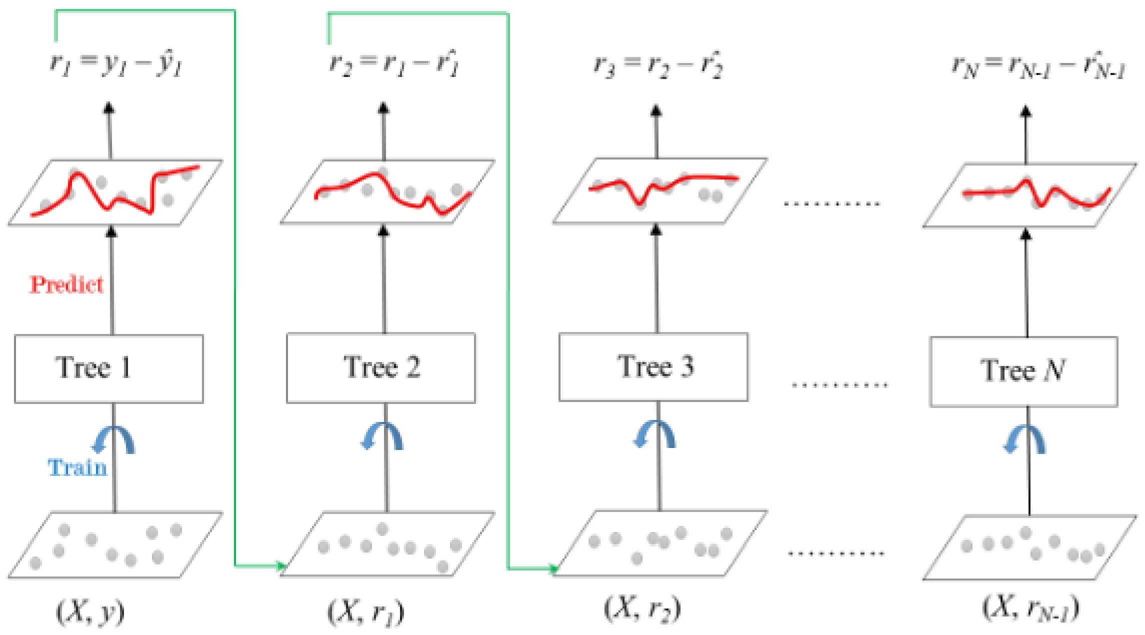


Fig 2. Working of Random Forest

Gradient Boosting :

Gradient Boosting is a popular boosting algorithm. In gradient boosting, each predictor corrects its predecessor's error. In contrast to Adaboost, the weights of the training instances are not tweaked, instead, each predictor is trained using the residual errors of predecessor as labels.
There is a technique called the **Gradient Boosted Trees** whose base learner is CART (Classification and Regression Trees).

The below diagram explains how gradient boosted trees are trained for regression problems.



Gradient Boosted Trees for Regression

The ensemble consists of *N* trees. Tree1 is trained using the feature matrix *X* and the labels *y*. The predictions labelled *y1(hat)* are used to determine the training set residual errors *r1*. Tree2 is then trained using the feature matrix *X* and the residual errors *r1* of Tree1 as labels. The predicted results *r1(hat)* are then used to determine the residual *r2*. The process is repeated until all the *N* trees forming the ensemble are trained.
There is an important parameter used in this technique known as **Shrinkage**.
Shrinkage refers to the fact that the prediction of each tree in the ensemble is shrunk after it is multiplied by the learning rate (eta) which ranges between 0 to 1. There is a trade-off between eta and number of estimators, decreasing learning rate needs to be compensated with increasing estimators in order to reach certain model performance. Since all trees are trained now, predictions can be made.
Each tree predicts a label and final prediction is given by the formula,

y(pred) = y1 + (eta * r1) + (eta * r2) + ... + (eta * rN)

<u>XGBoost</u>:

- XGBoost is an ensemble learning method. Sometimes, it may not be sufficient to rely upon the results of just one machine learning model.
- Ensemble learning offers a systematic solution to combine the predictive power of multiple learners. The resultant is a single model which gives the aggregated output from several models.
- 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.

Model performance:

- **Mean Squared Error:**

Mean Squared Error, or MSE for short, is a popular error metric for regression problems.
It is also an important loss function for algorithms fit or optimized using the least squares framing of a regression problem. Here “least squares” refers to minimizing the mean squared error between predictions and expected values.
The MSE is calculated as the mean or average of the squared differences between predicted and expected target values in a dataset.

MSE = 1 / N * sum for i to N (y_i – yhat_i) ^2.

Where *y_i* is the i’th expected value in the dataset and *yhat_i* is the i’th predicted value. The difference between these two values is squared, which has the effect of removing the sign, resulting in a positive error value.

R2 score:

R2 score is a metric that tells the performance of your model, not the loss in an absolute sense that how many wells did your model performs. In contrast, MAE and MSE depend on the context as we have seen whereas the R2 score is independent of context. So, with help of R squared we have a baseline model to compare a model which none of the other metrics provides. The same we have in classification problems which we call a threshold which is fixed at 0.5. So basically R2 squared calculates how must regression line is better than a mean line. But how to interpret R2 score. The normal case is when the R2 score is between zero and one like 0.8 which means your model is capable to explain 80 per cent of the variance of data.

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2}$$

Fig. Formula for Calculating R^2

Adjusted R-squared

For a multiple regression model, R-squared increases or remains the same as we add new predictors to the model, even if the newly added predictors are independent of the target variable and don't add any value to the predicting power of the model. Adjusted R-squared eliminates this drawback of R-squared. It only increases if the newly added predictor improves the model's predicting power. Adding independent and irrelevant predictors to a regression model results in a decrease in the adjusted R-squared.

$$R^2_{adj} = 1 - \frac{(1 - R^2)(n - 1)}{n - p - 1}$$

$where:$

$R^2 = R - squared$

$n = number\ of\ samples/rows\ in\ the\ data\ set$

$p = number\ of\ predictors/features$

Hyper parameter tuning:

Grid Search CV

Grid Search combines a selection of hyper parameters 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.

Conclusion:

The project comes to an end at this point. Beginning with loading the dataset, so far we have done EDA, pre-processing the data, Label encoding, Scaling the data, splitting the data into train and test data, applying various machine learning algorithms followed by hyper parameter tuning. We implemented 8 M.L. models. After comparing the mean square error and mean root square error of all the models, XGBoost has least mean square error and root mean square error. XGBoost has highest accuracy of 91.9% among all algorithms. So, We can conclude that XGBoost is the best model to predict rented bike count. The number of business hours of the day and the demand for rented bikes were most correlated and It makes sense also. Highest number of bike rented at the 18th hour of day. Total number of bike count increased when there was favourable temperature. So, this can be an important factor in predicting underlying patterns of rented bike count.