Supervised Machine Learning – Bike Sharing Demand Prediction

Sri harish A ,Sethupathy M

Capstone project 2 – Alma Better

Abstract:

The purpose of this project is to try a machine learning approach for predicting bike sharing demand in Seoul by given the hour, day, and information about the weather. The research contains: Data exploration, feature engineering, choosing appropriate scoring metric, cross algorithms, cross validation, tuning the algorithms, analysis of feature importance, analysis of residuals and performance evaluation. The used dataset is from years 2017 and 2018. With Gradient boost regressor the lowest RMSE and highest r2 score achieved on test set which is 9.234248 and 94%.

In this experiment by using EDA we are analyzing and visualizing the different perspectives. Which gives us different insights like

- Distribution of Independent and dependent variables.
- How does the Rented Bike count varies with season?
- How does the Rented Bike count varies with month?
- How does the Rented Bike count varies with holiday?
- How does the Rented Bike count varies with Functioning day?
- On which month, day and hour the rental bike provider floods with bike bookings.
- And also, we have used various Machine Learning algorithms to predict the rented bike count.

Problem Statement

This project contains the real-world data record of Rented Bike Demand of Seoul containing details like weather, holiday, functioning day, date etc. from 2017 to 2018. There are a total of 14 variables describing the 8800 observations. Each observation represents a Rented Bike Demand per hour.

Main aim of the project is to understand and visualize dataset from Bike renting firm point of view. The predictions of future bike per hour use could help for a better management of the service.

Features:

- **Date** - year-month-day.
- **Hour** - Hour of the day.
- **Temperature** - Temperature in Celsius.
- **Humidity** - Humidity in percentage.
- **Wind speed** - Wind speed in m/s.
- **Visibility** - Visibility in meters.
- **Dew point temperature** - The dew point is the temperature at which air is saturated with water vapor. (Celsius)
- **Solar radiation** - Solar radiation is the heat and light and other radiation given off by the Sun. (MJ/m2)
- **Rainfall** – Rainfall in mm.
- **Snowfall** – Snowfall in cm
- **Seasons** - Winter, Spring, Summer, Autumn
- **Holiday** - Holiday/No holiday
- **Functional Day** – No Func (Non-Functional Hours), Func (Functional hours)
- **Rented Bike count** - Count of bikes rented at each hour.

Introduction

The bike sharing is one of the methods of reducing city traffic. It is also lowering the air pollution by reducing the number of cars on the roads. The hypothesis in the research is that the bike sharing is highly related with the time of the day, season and weather conditions. The research will try to predict the bike shares in the future. The predictions of future use could help for a better management of the service. Another point of view is to test the machine learning algorithms how good are at solving this problem.

The first bike-share programs began in 1960s Europe, but the concept didn't take off worldwide until the mid-2000s. In North America, they tend to be affiliated with municipal governments, though some programs, particularly in small college towns, center on university campuses.

The typical bike-share has several defining characteristics and features, including station-based bikes and payment systems, membership and pass fees, and per-hour usage fees. Programs are generally intuitive enough for novice users to understand. And, despite some variation, the differences are usually small enough to prevent confusion when a regular user of one city's bike-share uses another city's program for the first time.

Benefits of Bike sharing demand prediction:

- Predicting Bike demands per hour is important for service providers to optimize bike allocation and station maintenance.
- By predicting right number of bikes helps to rebalance the bikes effectively which improves service quality.
- Unnecessary cancellations and waiting can be avoided.

Benefits of Bike sharing:

- **Environmental benefit:** Biking is great for the environment because it allows people to travel without releasing toxins and burning fuel.
- **Fitness:** Biking is known to be an excellent exercise for both the body and the mind.
- **Convenient to use:** Public bicycle systems offer a convenient, easy-to-use system for citizens.

Steps involved:

Outliers Treatment

Our dataset has many outliers which could affect the effectiveness of the model. So, by using bar plot outliers had been detected and by using Z- score technique all the outliers had been imputed with median.

Format conversion

As shown above, the problem is that in the data frame the day, month and year are given in separate columns; the date columns are read as an object type instead of a date type, which prevents it from accessing any date-related functionalities in Pandas. So we have created a function that integrates these columns and outputs date in a single column for convenience.

Exploratory Data Analysis

After loading the dataset, we performed this method by comparing our target variable that is 'Rented Bike Count' with other independent variables. This process helped us figure out various aspects and relationships among the target and the independent variables. It gave us a better idea of which feature behaves in which manner compared to the target variable.

Building structured multi-plot grids and graphs

When exploring multidimensional data, a useful approach is to draw multiple instances of the same plot on different subsets of your dataset. It allows a viewer to quickly extract a large amount of information about a complex dataset. Matplotlib offers good support for making figures with multiple axes; seaborn builds on top of this to directly link the structure of the plot to the structure of your dataset.

We have used count plot, box plot, distplot and pie chart in multi-plots for various features and the graphs provide a high level of convenience for comparison

Multicollinearity for all the variables has been plotted and some highly correlated variables have been detected.

Libraries that have been used

Pandas

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals.

Pandas is mainly used for data analysis and associated manipulation of tabular data in Data Frames. Pandas allows importing data from various file formats such as comma-separated values, JSON, Parquet, SQL database tables or queries, and Microsoft Excel. Pandas allows various data manipulation operations such as merging, reshaping, selecting, as well as data cleaning, and data wrangling features. The development of pandas introduced into Python many comparable features of working with Data frames that were established in the R programming language. The pandas library is built upon another library NumPy, which is oriented to efficiently working with arrays instead of the features of working on Data frames.

NumPy

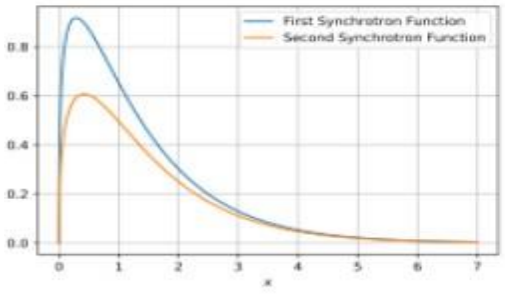
NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. NumPy targets the CPython reference implementation of Python, which is a non-optimizing bytecode interpreter. Mathematical algorithms written for this version of Python often run much slower than compiled equivalents due to the absence of compiler optimization. NumPy addresses the slowness problem partly by providing multidimensional arrays and functions and operators that operate efficiently on arrays; using these requires rewriting some code, mostly inner loops, using NumPy.

Using NumPy in Python gives functionality comparable to MATLAB since they are both interpreted, and they both allow the user to write fast programs as long as most operations work on arrays or matrices instead of scalars. In comparison, MATLAB boasts a large number of additional toolboxes, notably Simulink, whereas NumPy is intrinsically integrated with Python, a more modern and complete programming language. Moreover, complementary Python packages are available; SciPy is a library that adds more MATLAB-like functionality and Matplotlib is a plotting package that provides MATLAB-like plotting functionality. Internally, both MATLAB and NumPy rely on BLAS and LAPACK for efficient linear algebra computations.

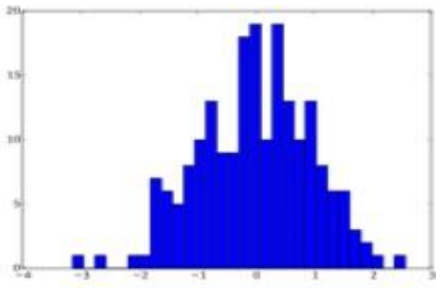
Python bindings of the widely used computer vision library OpenCV utilize NumPy arrays to store and operate on data. Since images with multiple channels are simply represented as three-dimensional arrays, indexing, slicing or masking with other arrays are very efficient ways to access specific pixels of an image. The NumPy array as a universal data structure in OpenCV for images, extracted feature points, filter kernels and many more vastly simplifies the programming workflow and debugging.

Matplotlib

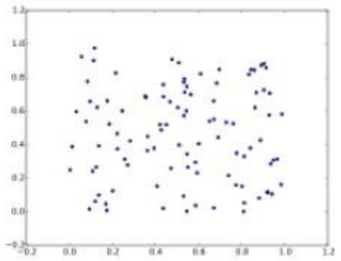
Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. SciPy makes use of Matplotlib.



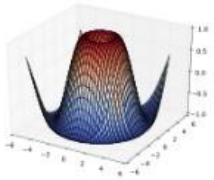
Line plot



Histogram



Scatter plot



3D plot

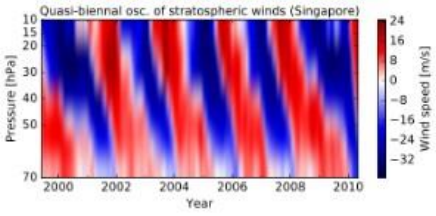
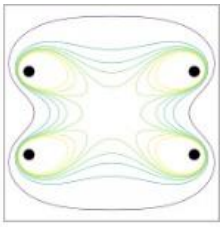
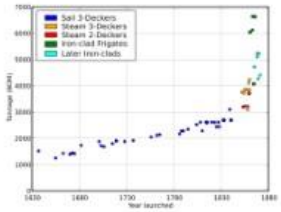


Image plot

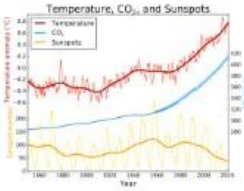


Contour plot

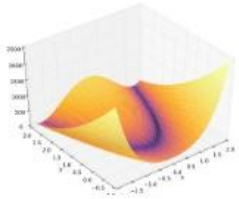


Scatter plot

Polar plot



Line plot



3-D plot

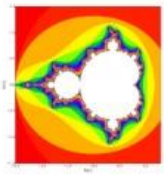
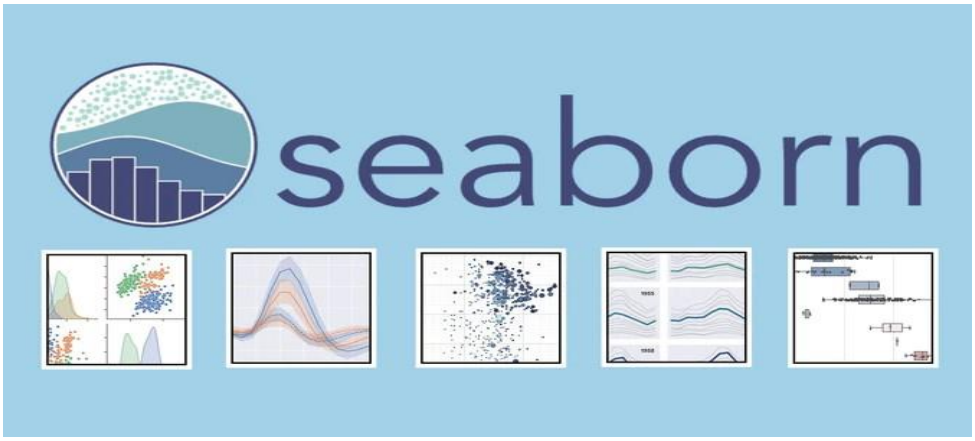


Image plot

Seaborn

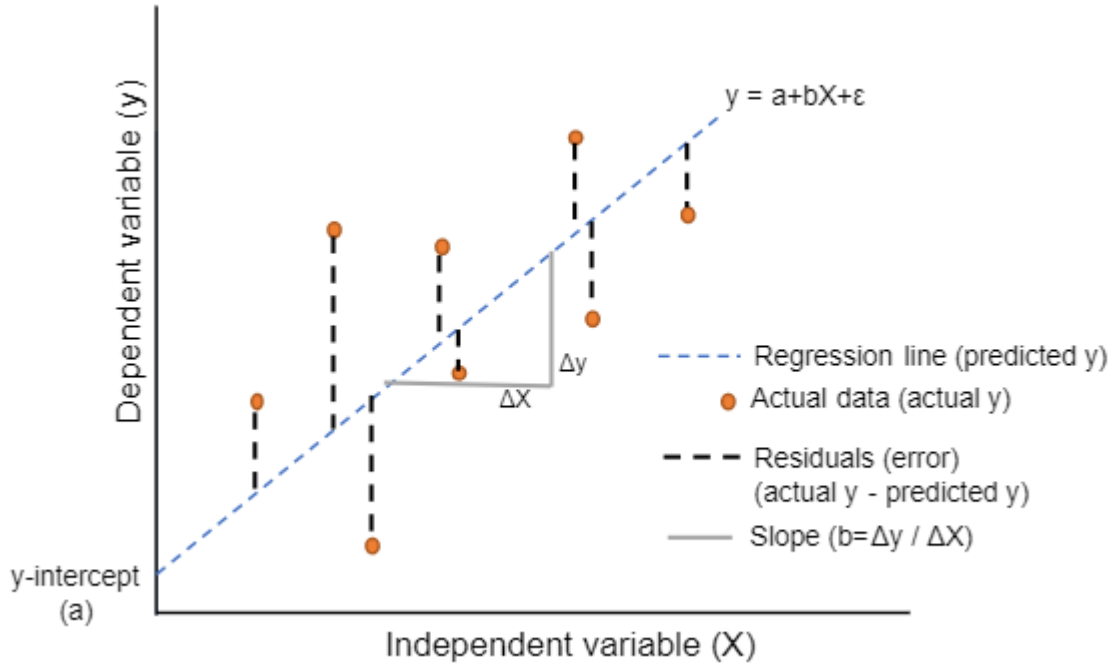
Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures.
Seaborn helps you explore and understand your data. Its plotting functions operate on data frames and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them. Behind the scenes, seaborn uses matplotlib to draw its plots



Machine Learning Models

Linear Regression

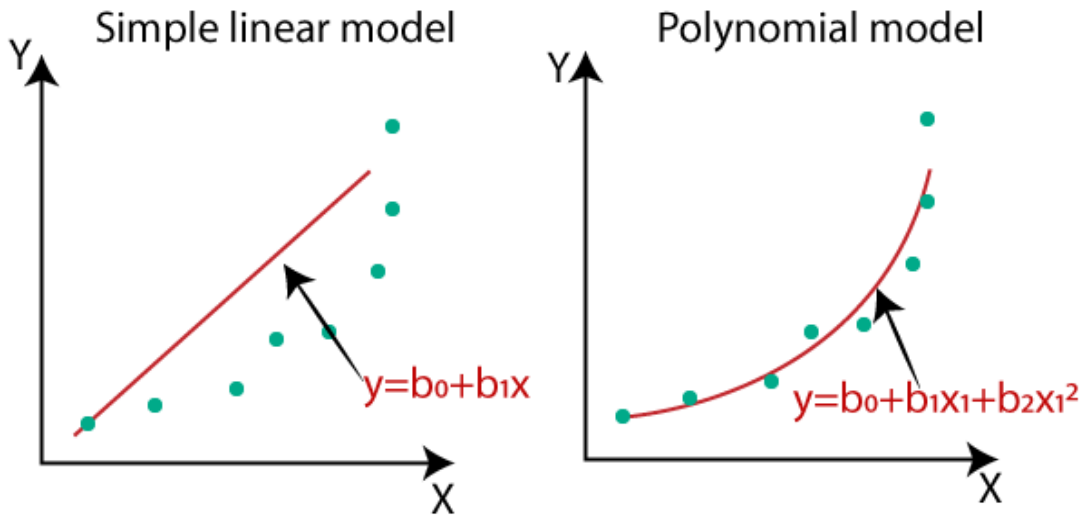
Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering, and the number of independent variables getting used.

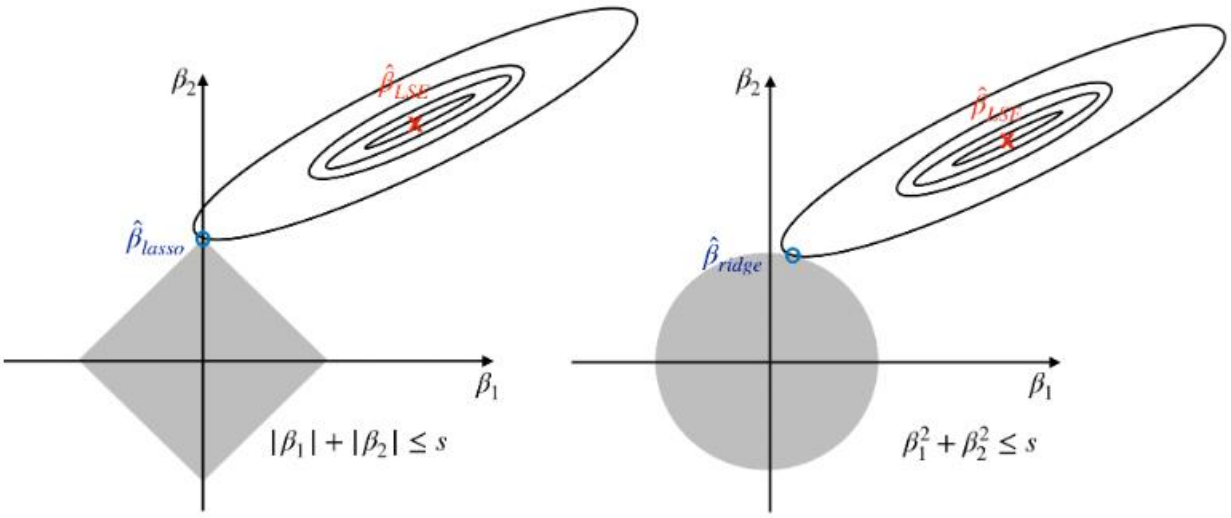


Polynomial Regression

Polynomial Regression is a regression algorithm that models the relationship between a dependent(y) and independent variable(x) as nth degree polynomial. It is also called the special case of Multiple Linear Regression in ML. Because we add some polynomial terms to the Multiple Linear regression equation to convert it into Polynomial Regression.

Lasso Regression





Lasso regression is a type of linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, like the mean. The lasso procedure encourages simple, sparse models (i.e., models with fewer parameters). This particular type of regression is well-suited for models showing high levels of multicollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination.
The acronym "LASSO" stands for Least Absolute Shrinkage and Selection Operator.

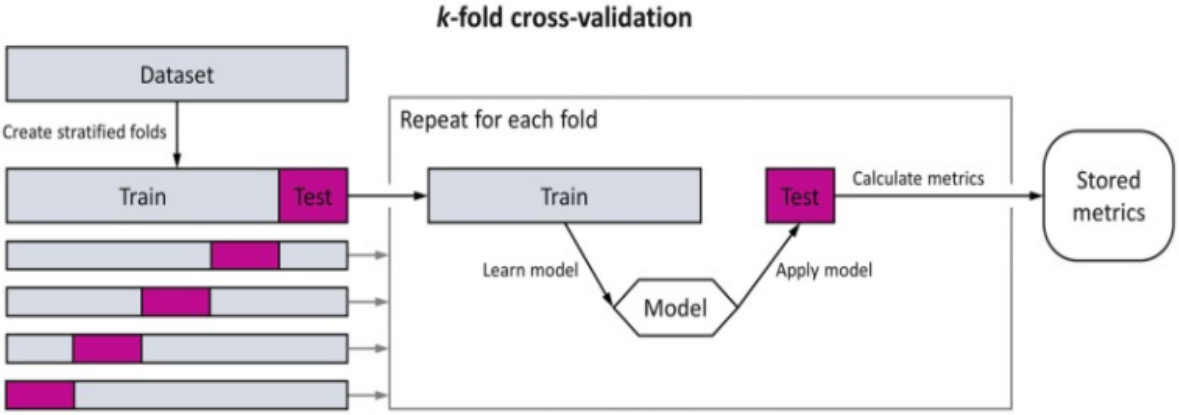
Ridge Regression

Ridge regression is a way to create a parsimonious model when the number of predictor variables in a set exceeds the number of observations, or when a data set has multicollinearity (correlations between predictor variables).
Tikhonov's method is basically the same as ridge regression, except that Tikhonov's has a larger set. It can produce solutions even when your data set contains a lot of statistical noise (unexplained variation in a sample).

Cross Validation

Cross-validation is a resampling method that uses different portions of the data to test and train a model on different iterations. It is mainly used in settings where

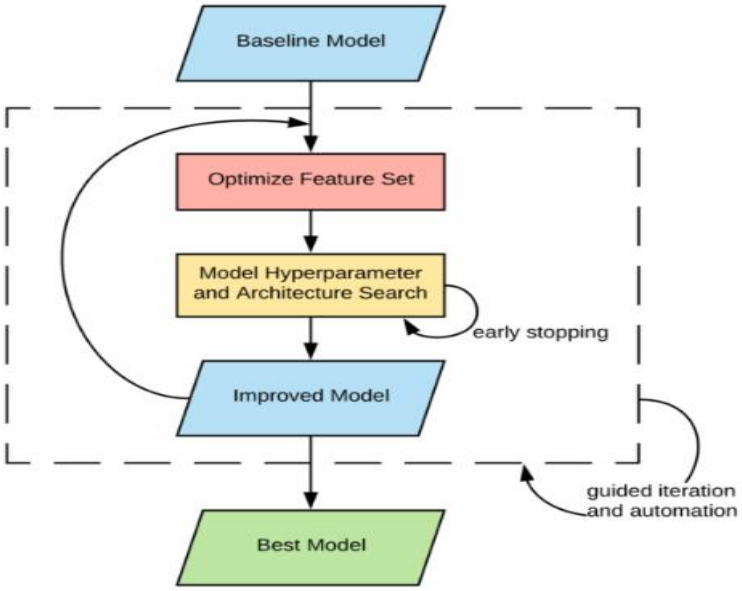
the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice.



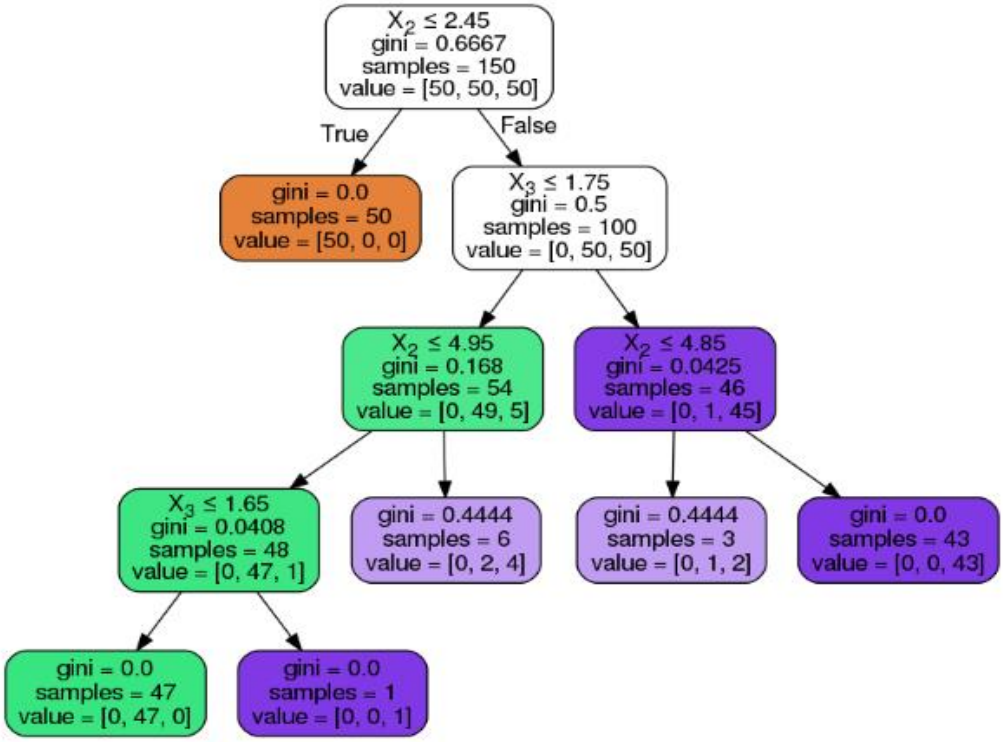
Hyperparameter Tuning

Hyperparameter tuning is the problem of choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a parameter whose value is used to control the learning process. By contrast, the values of other parameters (typically node weights) are learned.

The same kind of machine learning model can require different constraints, weights or learning rates to generalize different data patterns. These measures are called hyperparameters, and have to be tuned so that the model can optimally solve the machine learning problem.



Decision Tree

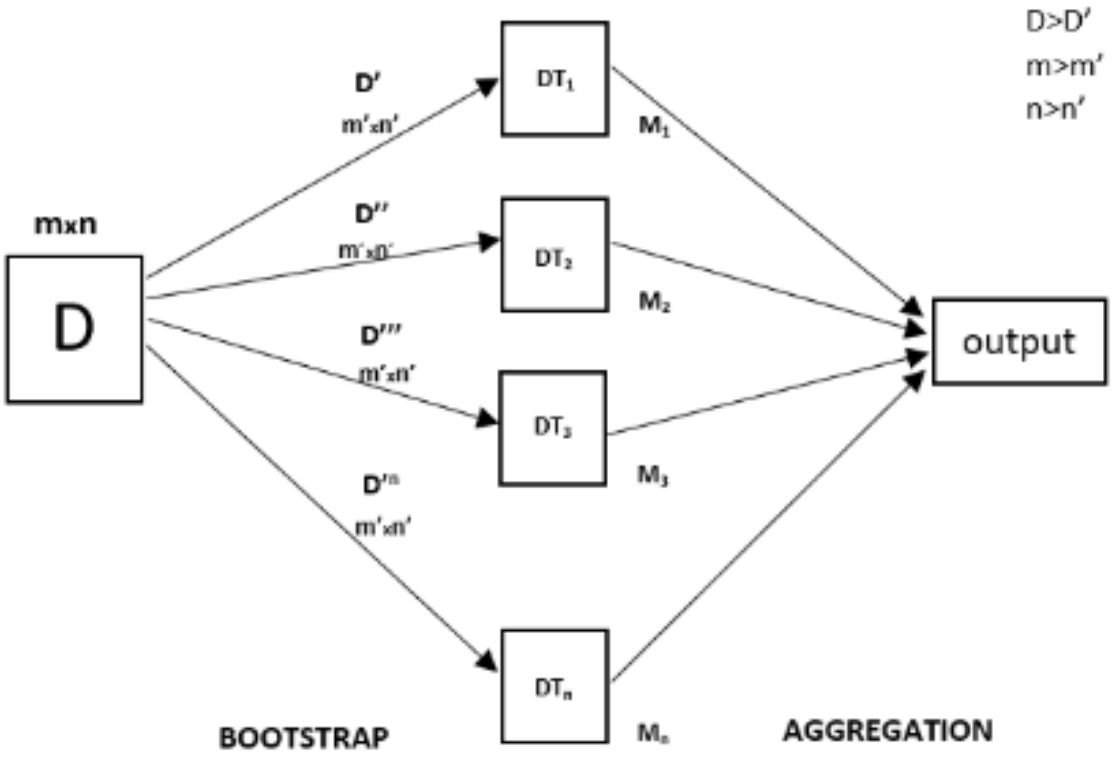


Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

A tree can be “learned” by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions. The construction of a decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle high-dimensional data. In general decision tree Regressor has good accuracy. Decision tree induction is a typical inductive approach to learn knowledge on classification and regression.

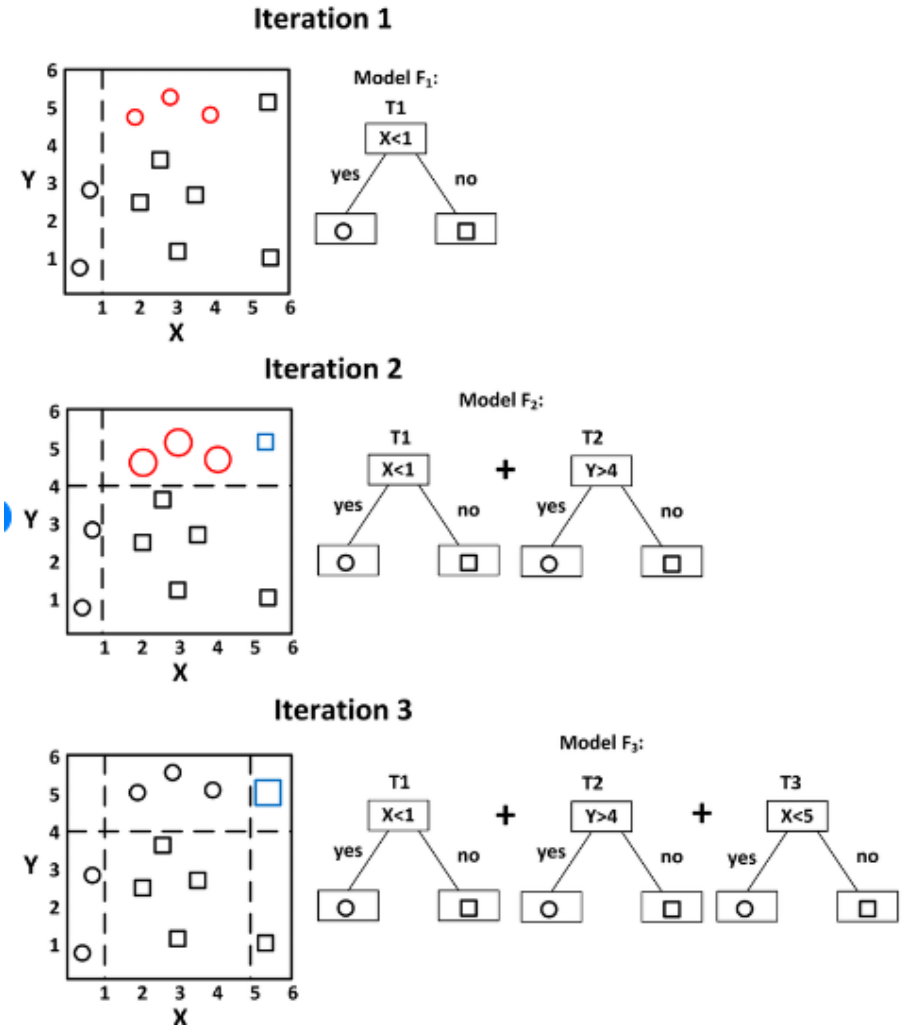
Random Forest Regressor

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 on 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 called Aggregation.



Gradient Boosting Regressor

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.[1][2] When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest.[1][2][3] A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.



Conclusion:

- It's quite obvious that most Bikes are rented during summer season.
- And the least number of bikes are rented during winter season.
- Hour of the day greatly influences the rented bike count.
- By using simple Polynomial Regressor algorithms we were able to get the **R2 score** of **90 percent.**
- Very little improvement in R2 score after using Lasso and Ridge with cross validation and hyper parameter tunning.
- By using simple Decision Tree algorithm, we couldn't get the desired results as over fitting occurred.
- The R2 score of Random forest regressor was **90** percent.
- We got the best **R2 score** of **94 percent** and low **RMSE** value of **9.23** from **Gradient Boost Regressor** after cross validation and hyper parameter tunning.
- Top five most important features are **Temperature, Humidity, Functioning day, Solar Radiation, Evening hour.**