**Seoul Bike Sharing Demand Prediction**

Abhishek Patil

**ABSTRACT**

Predicting the length of rental bike trips in the Seoul Bike sharing system is the goal of this investigation. Temperature, humidity, wind speed, visibility, dew point, solar radiation, snowfall, rain, number of bikes rented every hour, and date information are among the variables used.

Our study can shed light on the potential causes of such label classifications through feature selection, data analysis, and prediction using machine learning algorithms that take prior trends into consideration to get the right count.

**PROBLEM STATEMENT**

Currently Many big cities have adopted the use of rental bikes to improve mobility comfort. It is crucial to make the rental bikes accessible and available to the general public at the appropriate time since it reduces waiting. Eventually, maintaining a steady supply of rental bikes for the city emerges as a top priority. Predicting the number of bikes needed to maintain a steady supply of rental bikes at each hour's interval is essential.

The major goal is to create a predictive model that would enable them to estimate bike demand in advance. This will provide a consistent supply of bikes wherever they are needed.

**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 –Yes/No

**INTRODUCTION**

More than 1000 cities already have bike-sharing programmes, especially in major metropolitan areas like New York City, Paris, Washington, DC, London, Beijing, and Barcelona. Renting a bike is a faster option than walking for short distances. In addition, it is more pleasant and environmentally friendly than driving.

because to continued pollution, global warming, and the depletion of energy resources. Utilizing renewable energy that doesn't hurt the environment and can be recycled has become a major priority in many nations. One nation that has done this well is South Korea, where renting motorcycles in Seoul is one of their most popular services. But it is vital to have a budget in order to avoid any problems, such as long wait times.

**Major Factors Affecting Bike Demand**

* **• Rainfall:** People tend to use rented bikes quite frequently because they are convenient to rent from anywhere and drop off anywhere else, and they are inexpensive enough to rent daily. However, weather conditions like rainfall have a significant impact on the number of rentals; people avoid renting bikes in the rainy season. So it follows that the number of hired bikes is adversely associated with rainfall.
* **• Snowfall:** Similar to rainfall, snowfall has a negative impact on the number of rental bikes since it makes driving difficult on icy roads.
* **• Visibility:** When someone can't see clearly, it's normal for them to avoid driving, which has an impact on the number of rental bikes. Although there are not many of these in Seoul.
* **• Temperature:** It has been observed that individuals steer clear of hiring bikes in cold weather. The average temperature in Seoul ranges from 27 to 32 degrees Celsius. People therefore tend to enjoy the warmer weather, which has an impact on bike rental.
* **Working Day or Not:** Compared to an Off day, people rent bikes more on a working day. Reason behind this is being the same I.e. they can be easily rented from any place and can be dropped off any other place, cheap enough to rent daily.
* **Traffic:** Even though this isn't mentioned in data, traffic also supports renting bike count indirectly. If traffic is high or large people visiting nearby walk or rent a bike for purpose.

**STEPS INVOLVED**

In order to go ahead for data visualization upon key factors we need to go for certain extra steps before proceeding to the main segment. In this part we are going with the following steps:

1. Importing Analytical necessary library classes for future analysis.
2. Reading the csv data file from Google drive.
3. Setting figure size for future visualization.
4. Removing future warnings in seaborne plots.
5. Visualizing all the columns of the respective Data frame.
6. Viewing all data information
7. Checking the Unique values in the column ( if any)
8. Converting the data types to similar objects as the Analysis Demands.
9. Formatting the “size” column into a single column in the dataset.
10. Eradicating special characters from the dataset columns.

* **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 hypotheses and to check assumptions with the help ofsummary statistics and graphical representations. It gives us better idea of which feature behaves in which manner compared to target variable.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 figuring out various aspects and relationships among the target and the independent variables.

* **EXAMINING** **NULL** **VALUES**

The most critical thing from which we can draw some observations is Dataset, however data comes with unexpected values too i.e. sometimes it may be Null or missing in other words the space might be blank. Thus, at the time of analysing the first thing which we will do is to examine the null or missing values on the Dataset. It is the first step that will make the results “more” accurate &should be handled before it affects the performance of the models that predict the outcome.

* **ENCODING OF COLUMNS**

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.

* **STANDARDIZATION OF FEATURES**

Our main motive through this step 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 different algorithms to it.

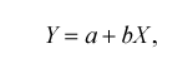
The basic goal was to enforce a level of consistency or uniformity to certain practices or operations within the selected environment.

* **FITTING DIFFERENT MODELS**
  + Linear Regression
  + Lasso Regression
  + Ridge Regression
  + K-Nearest Neighbour
  + Support Vector Machine
  + Decision Tree
  + Random Forest
  + Extra Trees
  + Gradient Boosting
  + Xgboost
  + LightGBM

**ALGORITHMS**

**I. Linear Regression**

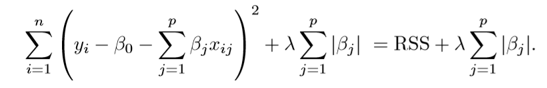
Linear regression (LR) is the simplest statistical regression method for identifying the linear link between the independent and the dependent variables. It is done by fitting a linear equation of line to the observed data. For fitting the model, it is utmost important to check, whether there is a connection between the variables or features of interest, which is supposed to use the numerical variable, that is the correlation coefficient. The following equation defines an LR line:

****

where X is the independent variable whereas Y is a dependent variable. b is the slope of the line and a is the intercept (the value of y when x = 0).

**II.** **Lasso Regression**

Lasso regression is a regularization technique. It is used over regression methods for a more accurate prediction. This model uses shrinkage. Shrinkage is where data values are shrunk towards a central point as 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. Lasso Regression uses L1 regularization technique.

****

This variation differs from ridge regression only in penalizing the high coefficients. It uses |βj| (modulus) instead of squares of β, as its penalty.

**III.** **Ridge Regression**

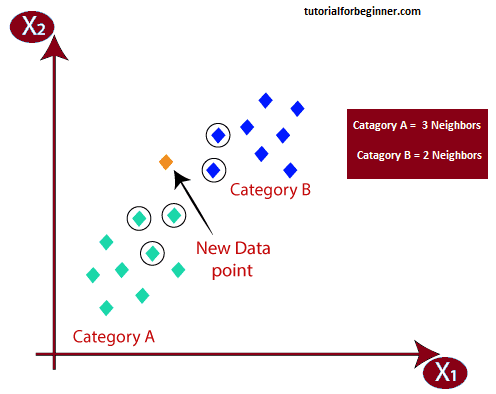
Ridge [regression](https://www.mygreatlearning.com/blog/what-is-regression/) is a model tuning method that is used to analyse any data that suffers from multicollinearity. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values being far away from the actual values.

Min(||Y – X(theta)||^2 + λ||theta||^2)

Lambda is the penalty term. λ given here is denoted by an alpha parameter in the ridge function. So, by changing the values of alpha, we are controlling the penalty term. The higher the values of alpha, the bigger is the penalty and therefore the magnitude of coefficients is reduced.

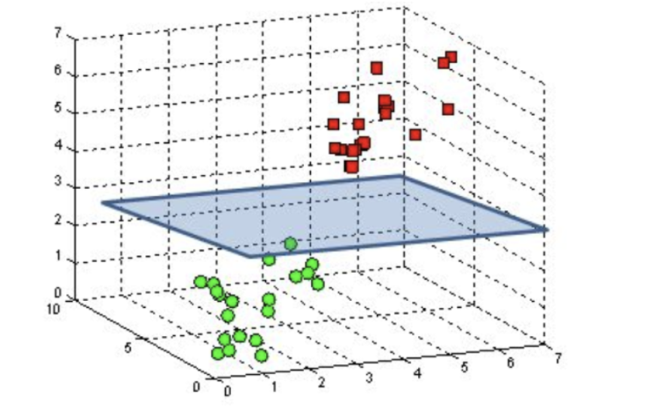
**IV. K-Nearest Neighbour**

KNN is a non-parametric supervised machine learning model which stores all the data available and predicts new cases based on a chosen similarity metric. The idea to predict the value of the new case based on the K nearest values available w.r.t to the similarity metric. It can be used for classification and regression problems. The common similarity metric used is Euclidian distance. Alternatively we can also use manhattan distance depending on the problem. The KNN model stores all the training data set and when a new test data point is given the aim is to find K points from the training set which are closest to the test data point. So here we are trying to find the points which minimise the similarity metric.

****

# ****V. Support Vertor Machine****

Support Vector Regression is a supervised learning algorithm that is used to predict discrete values. Support Vector Regression uses the same principle as the SVMs. The basic idea behind SVR is to find the best fit line. In SVR, the best fit line is the hyperplane that has the maximum number of points.

****

**VI. Decision Tree**

Decision Tree is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs and utility. Decision-tree algorithm fall under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables. The branches/edges represent the result of the node and the nodes have either:

1. Conditions [Decision Nodes]
2. Result [End Nodes]

The branches/edges represent the truth/falsity of the statement and makes a decision based on that in the example below which shows a decision tree.

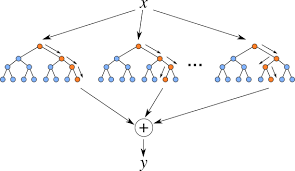
Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values.



**VII. 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 takes their majority vote for classification and average in case of regression

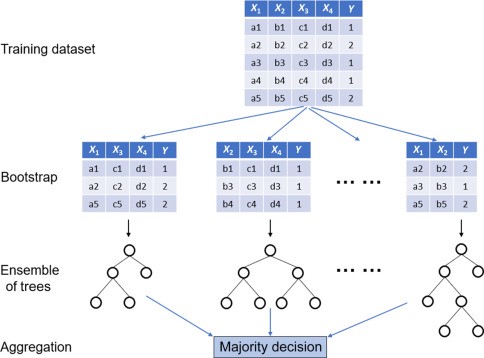
One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification.



**VIII. Extra Trees Regressor**

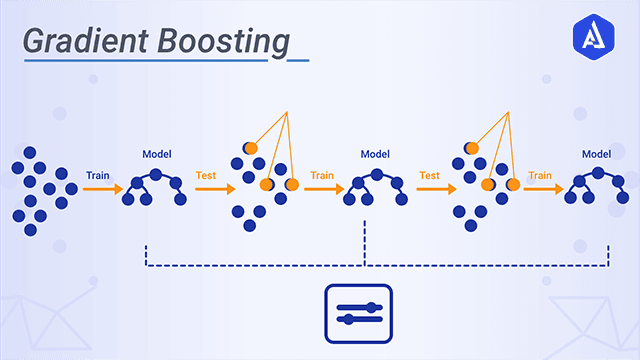
Extra trees (short for extremely randomized trees) are an ensemble supervised machine learning method that uses decision trees and is used by the Train Using AutoML tool. See [Decision trees classification and regression algorithm](https://pro.arcgis.com/en/pro-app/3.0/tool-reference/geoai/how-decision-tree-classification-and-regression-works.htm) for information about how decision trees work. This method is similar to [random forests](https://pro.arcgis.com/en/pro-app/3.0/tool-reference/geoai/how-random-forest-classification-and-regression-works.htm) but can be faster.

The extra trees algorithm, like the random forests algorithm, creates many decision trees, but the sampling for each tree is random, without replacement. This creates a dataset for each tree with unique samples. A specific number of features, from the total set of features, are also selected randomly for each tree. The most important and unique characteristic of extra trees is the random selection of a splitting value for a feature. Instead of calculating a locally optimal value using Gini or entropy to split the data, the algorithm randomly selects a split value. This makes the trees diversified and uncorrelated.



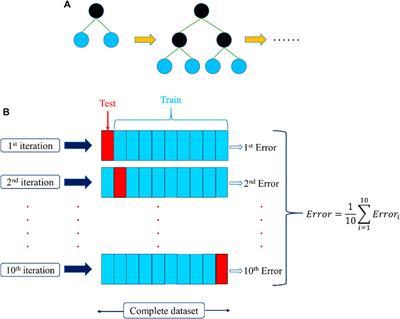
**IX. Gradient Boosting Regressor**

Gradient boosting is one of the most popular machine learning algorithms for tabular datasets. It is powerful enough to find any nonlinear relationship between your model target and features and has great usability that can deal with missing values, outliers, and high cardinality categorical values on your features without any special treatment. While you can build barebone gradient boosting trees using some popular libraries such as [XGBoost](https://xgboost.readthedocs.io/en/stable/" \t "_blank) or [LightGBM](https://lightgbm.readthedocs.io/en/latest/" \t "_blank) without knowing any details of the algorithm, you still want to know how it works when you start tuning hyper-parameters, customizing the loss functions, etc., to get better quality on your model.



# X. XGBoost Regressor

Extreme Gradient Boosting (XGBoost) is an open-source library that provides an efficient and effective implementation of the gradient boosting algorithm. Shortly after its development and initial release, XGBoost became the go-to method and often the key component in winning solutions for a range of problems in machine learning competitions. Regression predictive modeling problems involve predicting a numerical value such as a dollar amount or a height. **XGBoost** can be used directly for **regression predictive modeling**



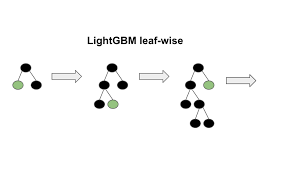
****

**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.

**XI. LightGBM**

**LightGBM** is a gradient boosting framework that uses tree based learning algorithms. It is designed to be distributed and efficient with the following advantages

* Faster training speed & higher efficiency.
* Lower memory usage.
* Better accuracy.
* Support of parallel, distributed, and GPU learning.
* Capable of handling large-scale data.



**MODEL PERFORMANCE**

**Mean Squared Error (MSE)**

MSE or Mean Squared Error is one of the most preferred metrics for regression tasks. It is simply the average of the squared difference between the target value and the value predicted by the regression model.

MSE = mean ((observed - predicted) ^2)

**Root Mean Squared Error (RMSE)**

which measures the average error performed by the model in predicting the outcome for an observation. Mathematically, the RMSE is the square root of the mean squared error (MSE), which is the average squared difference between the observed actual outcome values and the values predicted by the model.

RMSE = sqrt (MSE). The lower the RMSE, the better the model.

**Mean Absolute Error (MAE)**

like the RMSE, the MAE measures the prediction error. Mathematically, it is the average absolute difference between observed and predicted outcomes.

MAE = mean (abs (observed - predicted)). MAE is less sensitive to outliers compared to RMSE.

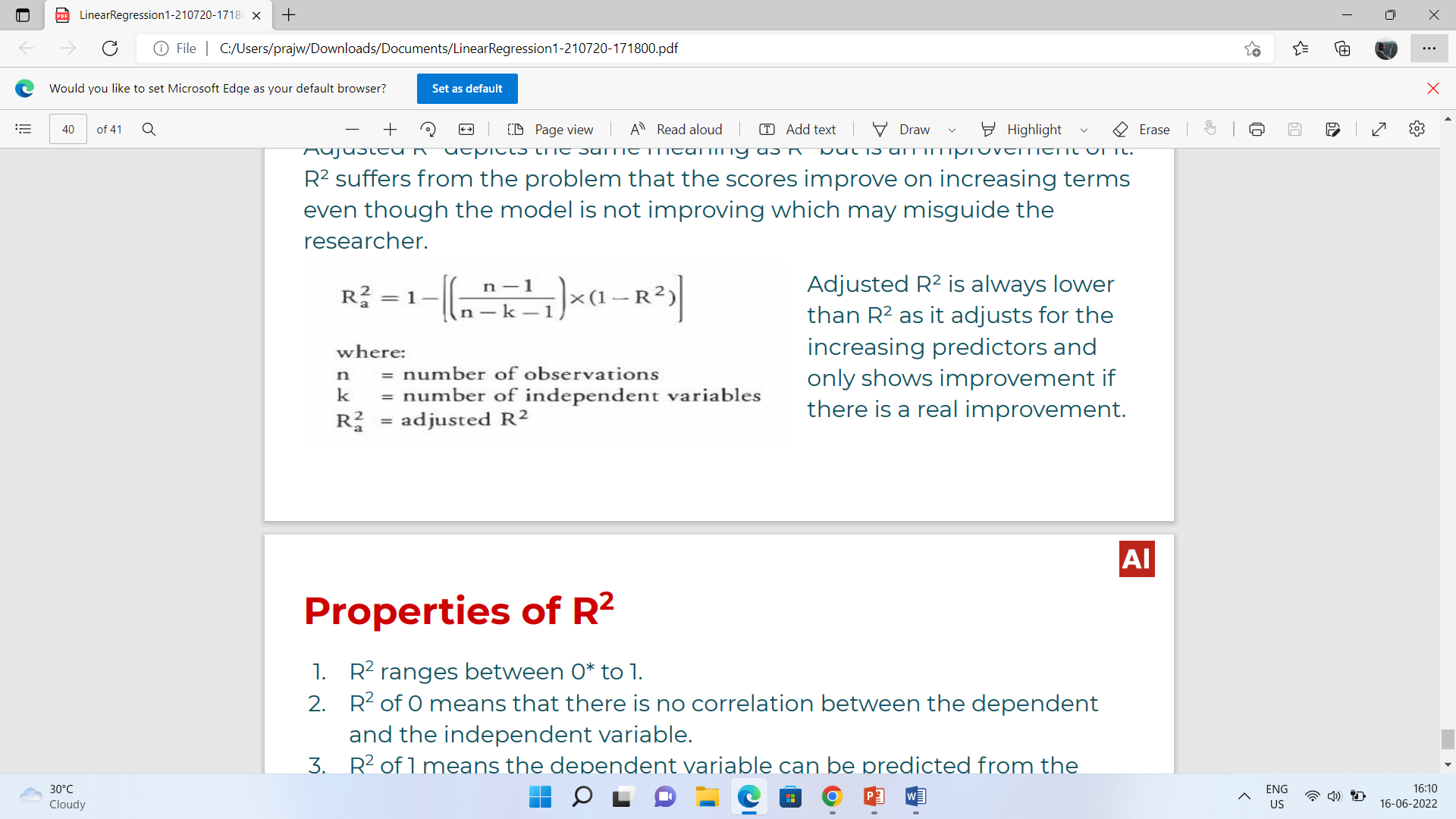
**R-squared (R²):**

which is the proportion of variation in the outcome that is explained by the predictor variables. In multiple regression models, R2 corresponds to the squared correlation between the observed outcome values and the predicted values by the model.

R² = 1- MSE(model)/MSE(Baseline). The Higher the R-squared, the better the model.

**Adjusted R²**

Adjusted R² depicts the same meaning as R² but is an improvement of it. R² suffers from the problem that the scores improve on increasing terms even though the model is not improving which may misguide the researcher.



Adjusted R² is always lower than R² as it adjusts for the increasing predictors and only shows improvement if there is a real improvement

**HYPER PARAMETER TUNING**

Hyperparameters are sets of information that are used to control the way of learning an algorithm. Their definitions 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, Randomized Search CV and Bayesian Optimization 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 Bayesian Optimization.

**Grid Search CV**

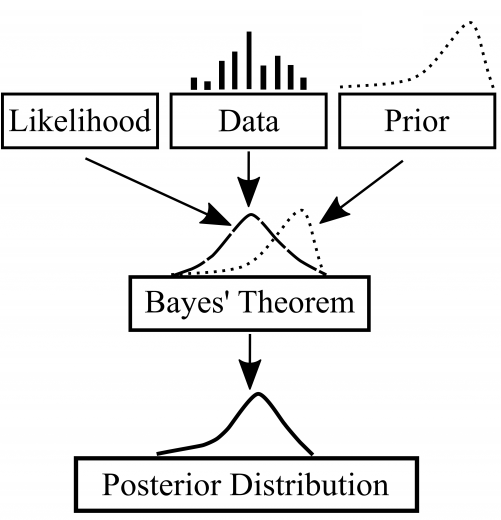
Grid Search combines a selection of hyperparameters established by the scientist and runs through all of them to evaluate the model’s performance. Its advantage is that it is a simple technique that will go through all the programmed combinations. The biggest disadvantage is that it traverses a specific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

**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

**Bayesian Optimization**

Bayesian hyper parameters optimization is a very efficient and interesting way to find good hyper parameters. In this approach, in naive interpretation way is to use a support model to find the best hyper parameters. A hyper parameter optimization process based on a probabilistic model, often Gaussian Process, will be used to find data from data observed in the later distribution of the performance of the given models or set of tested hyper parameters.



As it is a Bayesian process each iteration, the distribution of the model’s performance in relation to the hyperparameters used is evaluated and a new probability distribution is generated. With this distribution it is possible to make a more appropriate choice of the set of values that we will use so that our algorithm learns in the best possible way.

**CONCLUSION**

The data from Seoul Bike are used in the analysis. Four methods of regression The travel duration is predicted using linear regression, decision trees, XG boosting, and random forests. The results of this statistical data analysis in both an exploratory analysis and prediction methods are interesting.

The findings of the trial indicate that: • Summer saw the highest number of bike rentals, followed by autumn, spring, and winter. The busiest months for renting bikes are May through July, and December through February are the least popular months.

• The working class makes up the majority of customers in the bike rental industry. The EDA analysis shows that in Seoul, the demand for bikes is higher during the weekdays while people are at work.

• Temperatures of 20 to 30 degrees, evening hours of 4 to 8 hours, and humidity levels of 40% to 60% are the when the demand for bikes is at its highest, the most favourable conditions.

• The demand for rental bikes is significantly influenced by temperature, daytime, sun radiation, and humidity.

Since there was little correlation between features and labels, the linear model's prediction was poor. The LightGBM model, which has a r2 score of 0.902 and an RMSE of 199.4, produces the best predictions.