Machine Learning

Course Project Report (Regression)

(Final Draft)

Title of the project: Seoul Bike Sharing Demand Data Set
Student Name: Tarun Kumar Reddy
Email: tarun.g-25@scds.saiuniversity.edu.in

1. Introduction

- Bike sharing is a very innovative way of tackling traffic congestion by reducing the number of private vehicles on the road. They also enhance first and last-mile connectivity and improve the reliability of the public transport system in a city. Rental Bikes have been introduced in Seoul, South Korea, to enhance mobility and comfort for the citizens. In a city like Seoul, where two-wheelers are preferred over other forms of transportation due to their adaptability, the demand for rental bikes will be quite high. It is essential that rental bikes are available and accessible to the public at the right time since it lessens the waiting time. Therefore, providing the city with a stable supply of rental bikes becomes a significant concern. We aim to make a model that predicts the demand for rental bikes at each hour for a stable supply.
- Usually, the demand for bikes depends on the weather conditions on that particular day. So, we take different attributes of weather and the rented bike count on that particular day as our data and train the machine learning model.
- Since we want to predict a continuous value, we use regression analysis to build a model that takes weather conditions as input to predict the number of bikes rented at each hour.

2. Dataset and Features

- The dataset was sourced from Open Data Square Service, which is a municipal undertaking in Korea.
- The dataset is multivariate in nature and has a total of 8760 samples with 14 Attributes.

The fourteen attributes are:

- 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)

- Among the 14 features, 4 are categorical data types, and the rest are all numeric data types.

Exploratory Data Analysis:

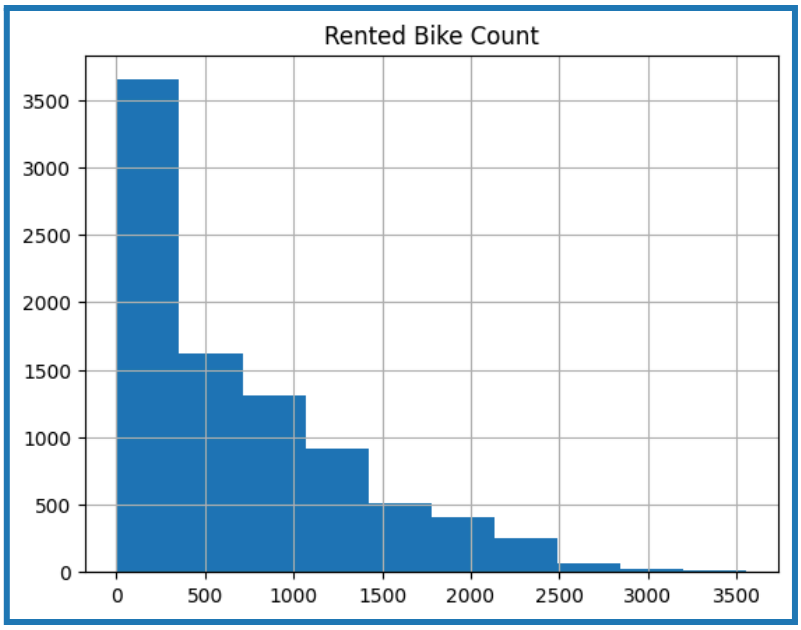
Performing Exploratory Data Analysis is a great way to understand the properties of the data set and its attributes, how they are correlated with each other, and to visualize the data.

- Started performing EDA by checking the null values in the data set, resulting in zero null values.
- Understanding descriptive statistics is a crucial thing in EDA, and it is performed using the pandas describe function.
- It is impossible to understand the data structure just by looking at the raw data, so we use different plots to help us understand it.

Rented Bike count:

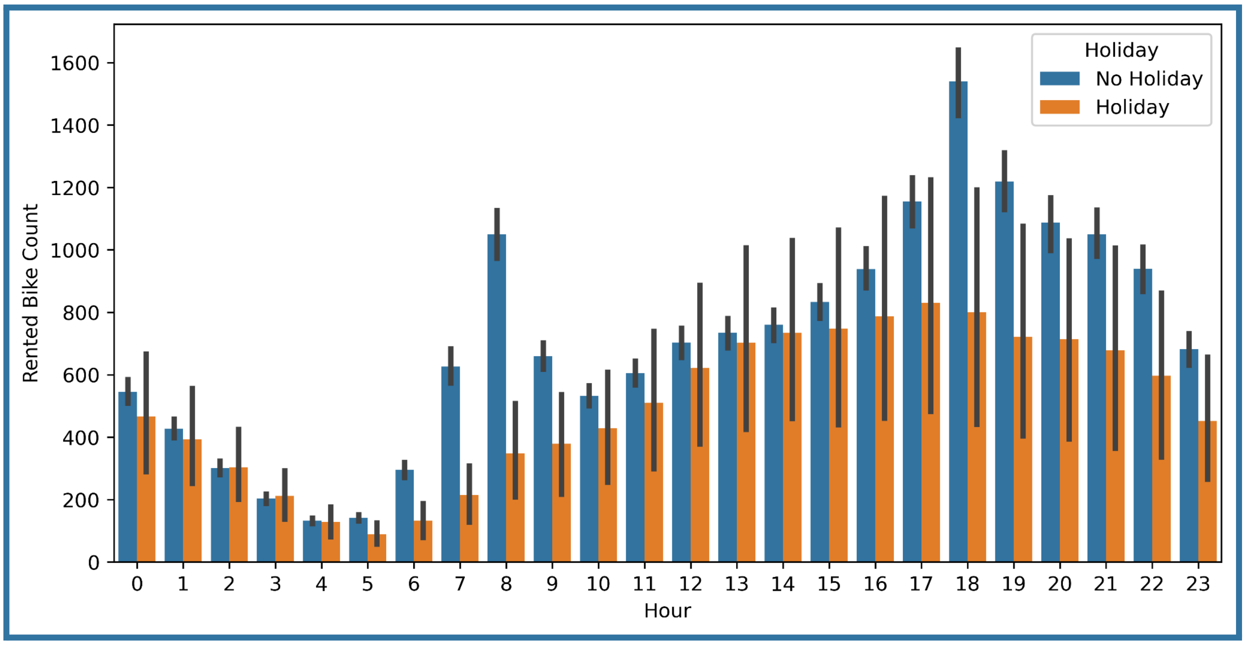
Rented Bike count is the dependent variable in the machine learning model.

It is a continuous variable, and hence, a histogram is the best way to visualize it. The histogram is right-skewed, which means most of the data falls in the initial bins. In this particular histogram, a huge amount of data falls into the first bin, and hence, the height of the bin is very high compared to other bins. From this histogram, we can interpret that most days, the count of rented bikes is around 0-500.



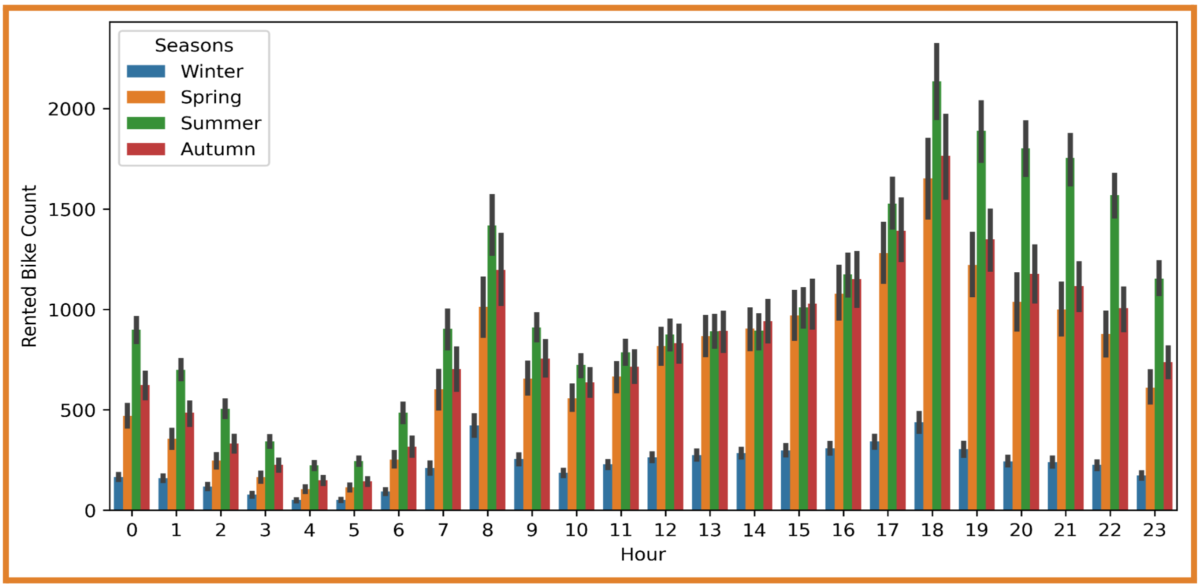
Rented Bike count and Holiday/No Holiday:

A double bar chart is plotted between the rented bike count and Holiday/No Holiday over the hours of the day. It is observed that there is more demand for bikes on No Holiday over Holiday days.



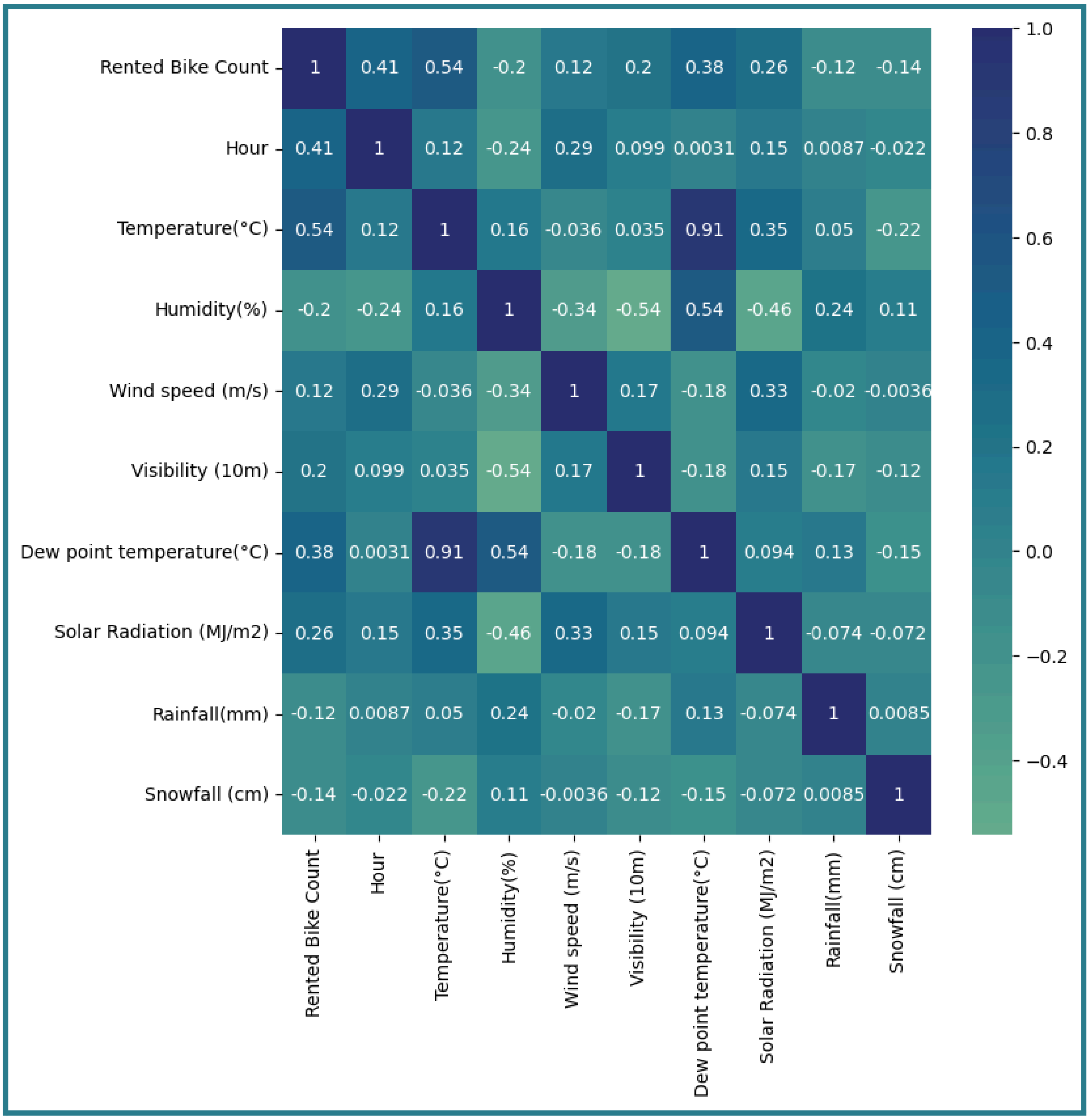
Rented Bike Count and Seasons:

A multi-bar chart plots rented bike count and seasons over the hours of a day. It observes that there is less demand for bikes in winter compared to other seasons. During summer, people may prefer to use more bikes due to the warm temperatures.



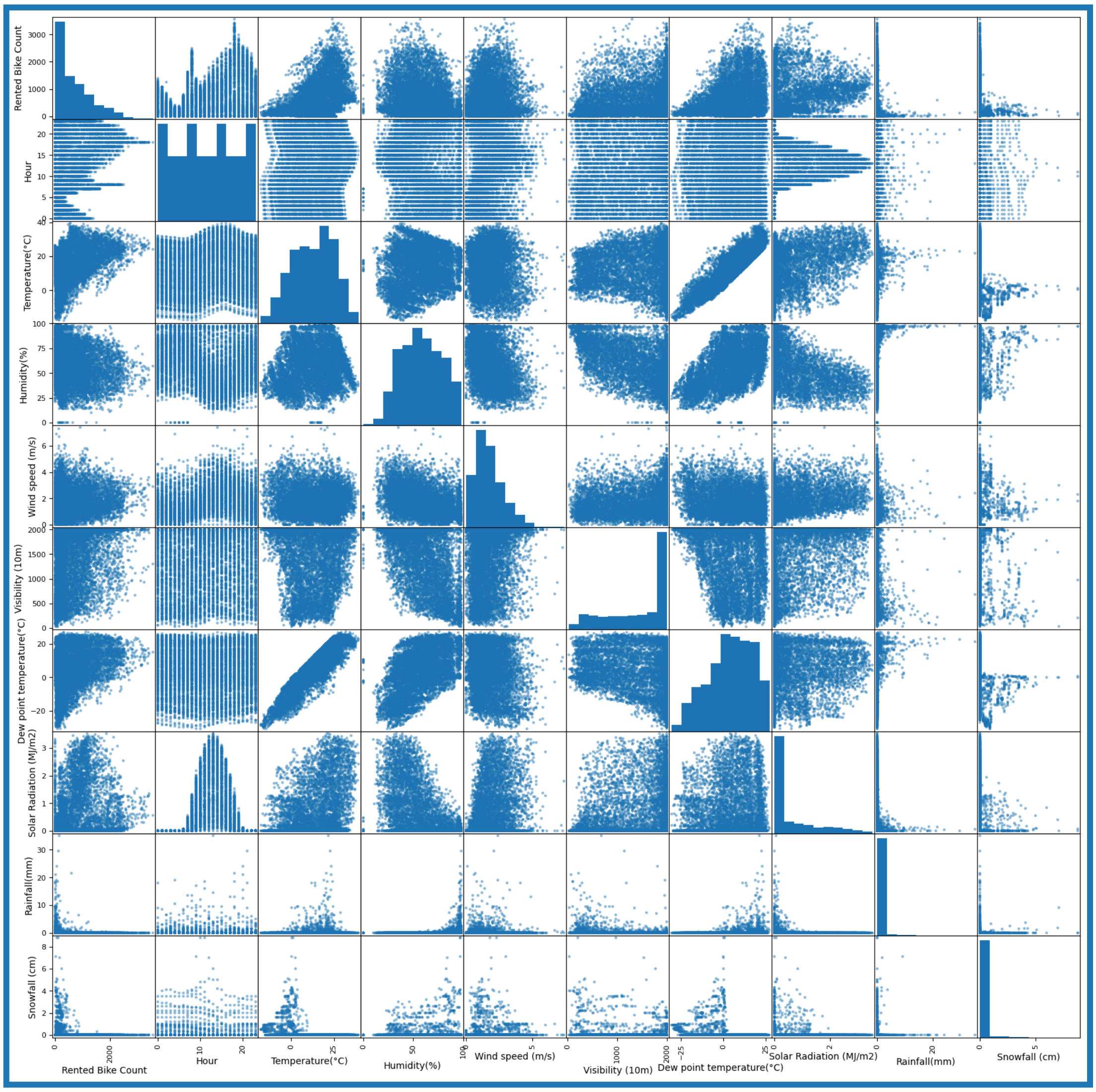
Correlation heat map:

Heat maps help us to visualise the intensity of a particular quantity. In the correlation heat map, we can see the correlation intensity between two variables and compare it with other correlations.



Scatter plot matrix:

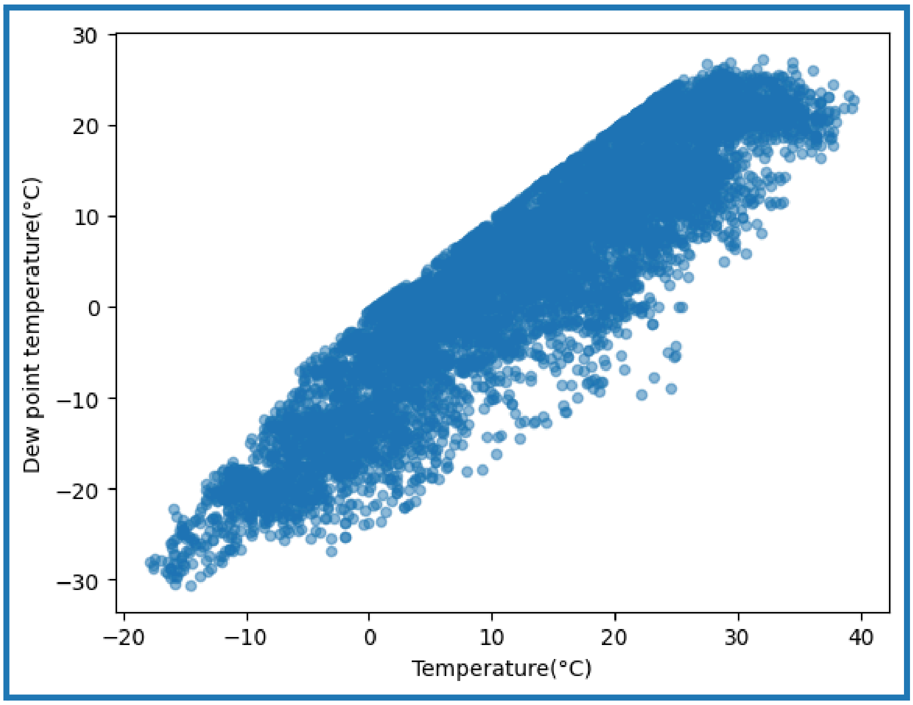
A scatter plot matrix is a matrix of scatter plots that shows the relationship between two variables in the whole data. It allows us to get a glimpse of how a feature is correlated with other features.



Correlation between Temperature and Dew point temperature.

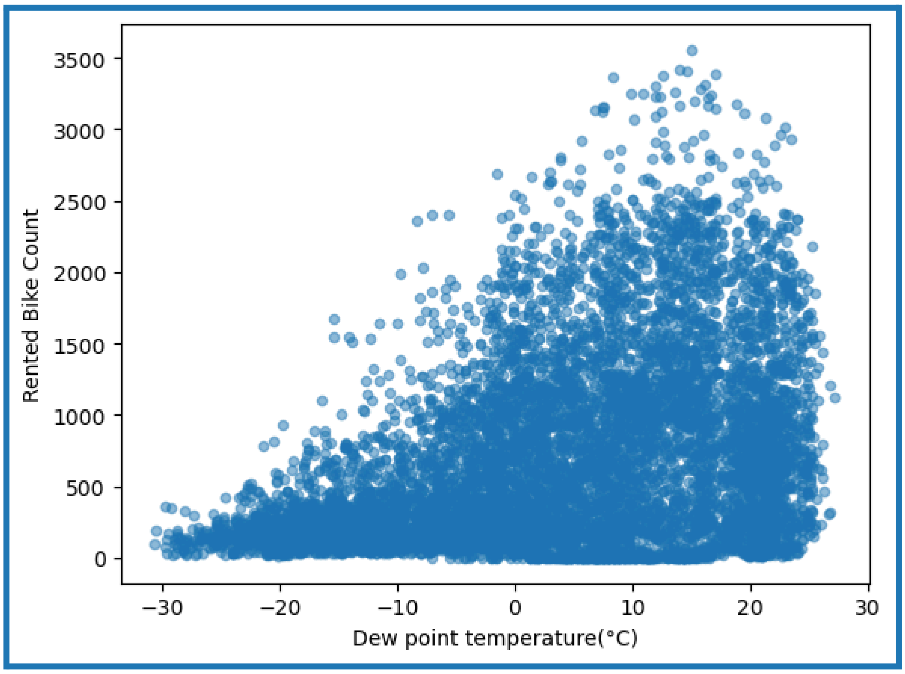
Based on the above analysis, there seems to be a very good correlation between Temperature and Dew point temperature. So, it is scaled as a whole graph to better understand it.

There is a perfect positive correlation between temperature and dewpoint temperature.



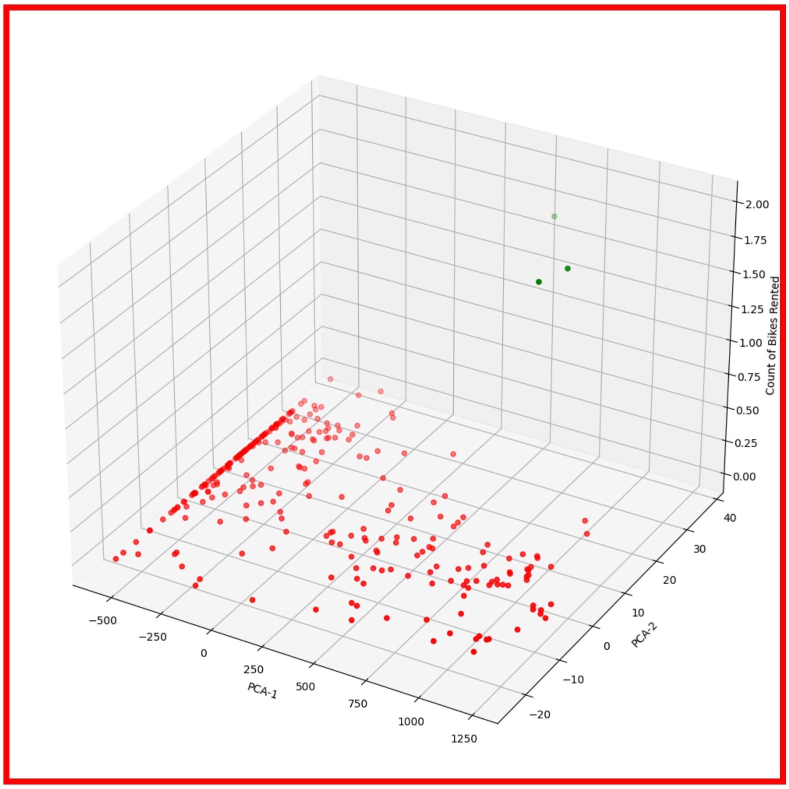
Correlation between Rented Bike Count and Dew point temperature.

Based on the above analysis, the dependent variable(Rented Bike Count) has a good correlation with dew point temperature. This means Dew Point Temperature plays a significant role in changing the demand for rented bikes.



Visualizing PCA:

Principal Component Analysis(PCA) is a statistical technique used to decompose higher dimensional data into lower dimensions by preserving the maximum amount of information in the data. Here, the 14 features were reduced to 2 features and those two features were plotted against the target variable, Rented Bike Count.



3. Methods

Following are the various methods used in this project.

3.1 Baseline - Linear Regression

- Linear Regression is a process of finding a linear relationship between independent and dependent features.
- Linear Regression Equation: $\hat{Y}=b_0+b_1X_1$
- It is one of the most popular ways to find patterns and predict a continuous value.
- This Linear Regression in Python can be achieved using a class called LinearRegression() in the Sklearn library

3.2 Polynomial regression

- Polynomial Regression is a process of finding non-linear relationships between independent and dependent features
- It finds the parameters for the n degree of the polynomial equation
- Polynomial Regression Equation:

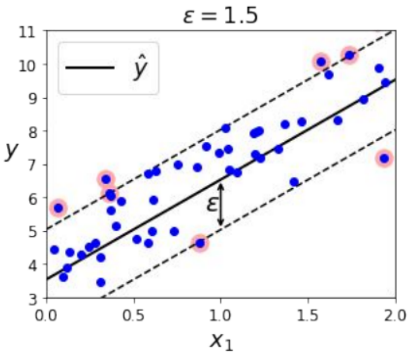
  $$Y = b_0+b_1x_1+ b_2x_1^2+ b_2x_1^3+\ldots\ldots b_nx_1^n$$
- Polynomial Regression can be developed in Python using a class called PolynomialFeatures from the Sklearn library

3.2 Regularization

- Regularization is a strategy being employed to overcome the overfitting of data in Polynomial Regression. This can be achieved by adding some weight to the loss function, and it can be achieved in three ways: LASSO Regression, Ridge Regression, and Elastic Net.
- L1 / LASSO Regression
  - (LASSO: **L**east **A**bsolute **S**hrinkage and **S**election **O**perator)
  - $J(\theta)=MSE(\theta)+2\boldsymbol{\alpha}\boldsymbol{\Sigma}^{n}{}_{i=1}\,abs(\theta_i)$
  - Eliminates least important features by setting the weight to zero
- L2 / Ridge Regression
  - $J(\theta)=MSE(\theta)+(\boldsymbol{\alpha}/m)\boldsymbol{\Sigma}^{n}{}_{i=1}\,\theta^2_i$
  - Ridge Regression keeps the model weights as small as possible
- ElasticNet Regression
  - It is the weighted Sum of L1 and L2
  - $J(\theta)=MSE(\theta)+r(2\boldsymbol{\alpha}\boldsymbol{\Sigma}^{n}{}_{i=1}\,abs(\theta_i))+(1-r)((\boldsymbol{\alpha}/m)\boldsymbol{\Sigma}^{n}{}_{i=1}\,\theta^2_i)$
  - Where $\boldsymbol{\alpha}$ and r are hyperparameters
- Sklearn provides different classes for all these three methods, and we can train the models using the same classes

3.2 Support Vector Machine (SVM) Regression

- SVM is one of the most robust approaches in Machine Learning, which can be used both in Regression and Classification
  - SVMs capable of performing both linear and non-linear regression
  - They use extreme data points as support vectors to make decision boundaries and make a hyperplane in the middle of those boundaries. (Shown in the fig below)



- Linear SVM Regression:
  - Linear SVMs try to find the hyperplane that best represents the point cloud.

- Kernel SVM Regression:
  Kernel SVMs transform the data into higher dimensions and then find a hyperplane that best represents the point cloud. It can be achieved in two ways:
  - Polynomial kernel
    - It will transform the data into higher dimensions by converting the data into polynomial features.

$$\phi(x_1, x_2) = (z_1, z_2, z_3) = (x_1, x_2, x_1^2 + x_2^2)$$

- Radial Basis Function kernel
  - Rbf is the widely used Kernel.
  - It is also called the Gaussian Kernel since it uses the Gaussian function to transform the data into higher dimensions.

    $$K(X_1, X_2) = \exp(-\frac{\|X_1 - X_2\|^2}{2\sigma^2})$$

  - The two parameters used in the above equation are $\mathbf{X}_{i,}$ the **mean,** and $\boldsymbol{\sigma}$, the **standard deviation.**

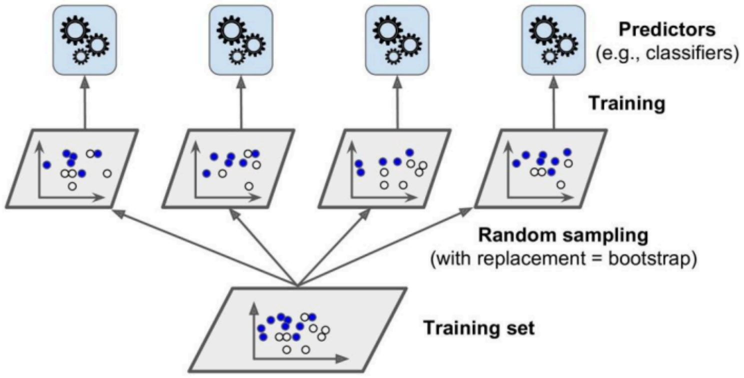
3.3 Decision Tree Regressor

- Decision Trees are inspired by symbolic AI.
- Decision Tree is a binary tree with a root node, internal nodes, and leaf nodes.
- This tree starts from the root, traverse through the internal nodes, and reaches the leaf nodes.
- It traverses along the way based on the threshold of that node's feature(s). The threshold will be obtained by the **CART algorithm**.
- The CART (Classification and Regression Trees) algorithm trains the Trees. It uses the **Mean Squared Error** of the data for the given threshold in the particular node to classify the data and move along the tree. It determines the threshold by minimizing the Mean Squared Error.
- The CART algorithm starts by splitting the Dataset into two based on a single feature, ***K*** and a threshold $\mathbf{T}_k$.
- (***K*** and $\boldsymbol{T_k}$) are chosen to produce the purest subset. This process repeats recursively until the given max_depth or mean squared error becomes zero.

3.4 Random Forest Regressor

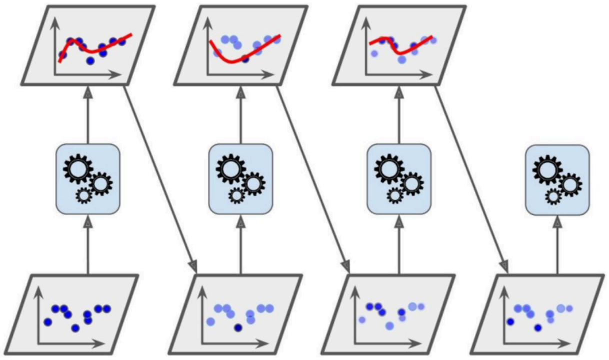
- Random Forest Regressor is an Ensemble learning technique.
- The ensemble learning algorithm trains different models and takes the average of all the models.
- In classification, the average is calculated using the statistical measure **mode,** whereas in regression, the **mean** is used as the average value.
- It splits the data set into different random subsets(With replacement) of equal size and trains different models on the subsets. It is also called **Bagging(Bootstrap Aggregating)**
- Training Decision Trees using bagging is called **Random Forest**

- Random Forest Regressor is a Random Forest technique used for regression.

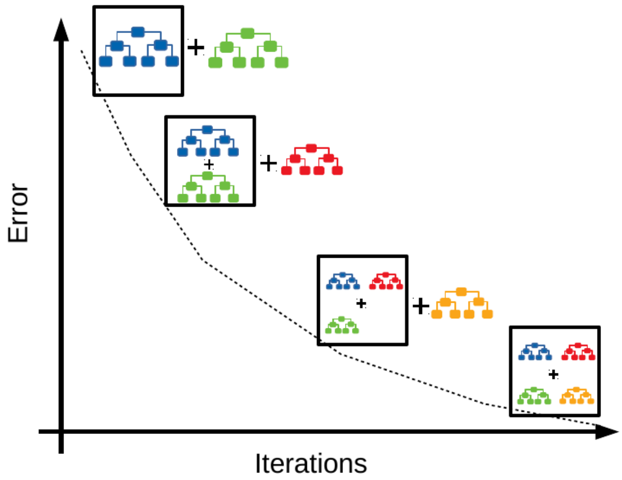


3.4 Boosting Regressor

- Boosting Regressor is an Ensemble learning technique.
- In the ensemble learning technique, we train different models and aggregate all the models to get the desired output.
- Boosting combines several weak learners to produce a strong learner.
- Boosting has two techniques.
  - Ada Boost (Adaptive Boosting)
  - Gradient boosting
- Ada boost combines different underfitted models and corrects them to make a new predictor. This will lead the new predictor to concentrate more on difficult scenarios.



- In the Gradient boosting technique, a new predictor will be obtained by correcting the residual in the previous predictors.



4. Experiments & Results

4.1 Protocol

Converting Date into time series data:

- The given data has a feature named date that we can convert that data into pandas time series data type. We can achieve this by the `pd.to_datetime()` method.
- It is further split into individual features of Day, Month, and Year.
- Doing this increased the accuracy of the models.

Encoding:

- Converting the categorical data into numeric data is called Encoding.
- There are different types of encoding techniques available. Ordinal Encoding is the best choice when it comes to encoding the training data set.
- **Seasons, Holiday, and Functioning Day** are the three categorical features converted into numeric data.

Feature Scaling:

- Feature scaling is done on the features to improve accuracy, reduce noise, and make the models converge quickly.
- In this particular project, Power Transformation and Standard Scaling were used.

Power Transformation:

- Power transformation is applied to data to lessen the skewness of the data and outliers.
- In this particular project, the Yeo-Johnson transformation is used:

$$y = \begin{cases} ((y+1)^\lambda - 1)/\lambda & \text{if } \lambda \neq 0, y \geq 0 \\ \log(y+1) & \text{if } \lambda = 0, y \geq 0 \\ -[(-y+1)^{(2-\lambda)} - 1]/(2-\lambda) & \text{if } \lambda \neq 2, y < 0 \\ -\log(-y+1) & \text{if } \lambda = 2, y < 0 \end{cases}$$

Every data point in the data set goes through the above transformation.

- This is performed only on the numeric data.

- This Yeo-Johnson method handles both negative and positive values.
- This transformation increased accuracy at a significant level.

Standard Scaling:

- To increase the performance of the Gradient Descent Algorithm and the efficiency of the model, we perform standard scaling (Standardisation) on the data set.
- Standardisation scales the data into a particular range(Unit Normal Distribution). It is done by subtracting every data point from its mean and dividing it by the standard deviation
  Standardisation: $\mathbf{X}' = (\mathbf{X} - \mu) / \sigma$
- We use the StandardScaler class from Scikit Learn to perform standardisation.

Feature Selection:

- We employ various feature selection techniques to reduce the number of input variables, thereby eliminating the noise present in the data while utilising more relevant data.
- We employ techniques such as Variance Threshold, Select K Best, and Select Percentile Feature Selection methods.
  - Variance Threshold removes all features whose variance is less than a threshold.
  - In Select K best, user-specified k best features are retained. F_regression helps estimate the degree of linear dependency between two features and the target variable.
  - Select Percentile Retains user-specified highest scoring percentage of features. By default, it always takes 10 Percent of features to keep.
- We use the feature_selection module from Scikit to learn how to perform feature selection.

Remarks:

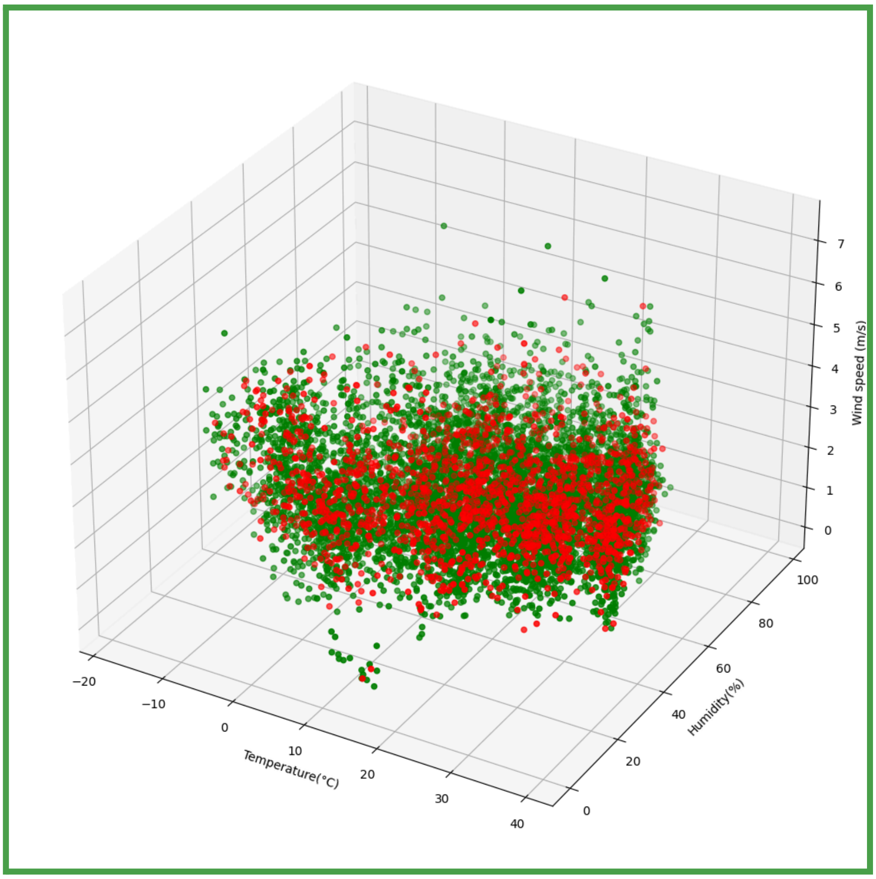
Although Feature selection is considered to improve accuracy, it doesn’t seem to improve the accuracy of this data here, rather decreasing it at certain places. The variance threshold looks pretty similar to the accuracy scores of non-feature selection. So, all 14 attributes are included in model training models.

Multicollinearity:

- Regression models have the underlying assumption that all the independent variables of the data are independent of each other. If they are dependent on each other, the results will not be good.
- In the given data, Temperature and Dew Point Temperature are highly correlated, with a correlation coefficient of **0.912343.** So, removing the Dew Point Temperature from the data improved the accuracy of the models.

Train-Test split

- The dataset was divided into two subsets, namely the training and the testing data, with a proportion of 80% and 20%, respectively.
- The training data are used to train the model, and the testing data can be used to evaluate its performance.
- This can be achieved using the built-in function in the Scikit learn library.
- We can use any three variables to visualise the data split. Here, it is visualised using Temperature(°C), Humidity(%), Wind speed (m/s).



4.2 Results

- Different models are trained using linear Regression, Polynomial Regression, LASSO Regression, Ridge Regression, Elastic Net, linear SVMs, and Kernel SVMs.
- The model can be evaluated using the ANOVA techniques by calculating the R-squared.
- The R-squared can be calculated as:

$$R^2 = 1 - \frac{RSS}{TSS}$$

Where:
RSS = Residual sum of squares
TSS = Total sum of squares

Linear Regression:

- The R-squared value turned out to be `0.5473867855588497`
- The R-squared value is the same even if we train the model with standardisation.
- We perform cross-validation to check the sensitivity of the model. Here, 10 10-fold cross-validation is performed using the Scikit learn library.
- Cross-validation report:
  `0.565695570912433 +/- 0.018314632343610502`
- The R-squared and cross-validation scores show that the linear regression technique is not efficient for this problem since its accuracy is pretty low.

Polynomial Regression:

- For the given problem, Polynomial Regression with degree 2 gives the maximum accuracy compared to other degrees, which is **0.5483926871573729**
- The R-squared value improved with standard scaling and gave an accuracy of **0.5674320849008825**
- Cross-validation report:
  0.565695570912433 +/- 0.018314632343610502
- This is also not a robust model since the accuracy of the model is low.

LASSO Regression:

- LASSO Regression gives an R-squared value of 0.5474048249724595
- The accuracy remains the same even after doing the standard scaling. The resulting accuracy of the model after standard scaling is **0.547477679730227**
- This accuracy could be a better score; hence, it is not a good model for this problem.
- Cross-validation report:
  0.5656946279604732 +/- 0.01827394289560667

Ridge Regression:

- Ridge Regression gives an R-squared value of **0.5473869488866467**
- Which is not a good score for the given regression problem.
- Performance is still the same even after standard scaling.
- Cross-validation report:
  0.5656956181278698 +/- 0.01831411205220463

Elastic Net:

- The Elastic Net approach gives an R-squared value of **0.5457862644215118; again**, this is not a good score.
- Performance is the same even after performing standard scaling.
- Cross-validation report:
  0.5656873672830806 +/- 0.018158121725113555

Linear Support Vector Machines:

- Linear Support Vector Machines gives an R-squared value of **0.10585998679227937**
- This R-squared value is the lowest compared to all other techniques.
- There is a drop in the performance after performing Standard Scaling on the data with an R-squared value of **-0.007484114071456993**
- Cross-validation report:
  0.1348449204990502 +/- 0.15704580161109005

Kernel Support Vector Machines:

Radial Basis Function(RBF) kernel:

- Kernel Support Vector Machines with RBF kernel give an R-squared value of **0.24025324909590007**
- The above score was increased to **0.2488501626525342** with standard scaling. Both the scores are very low compared to the other models.

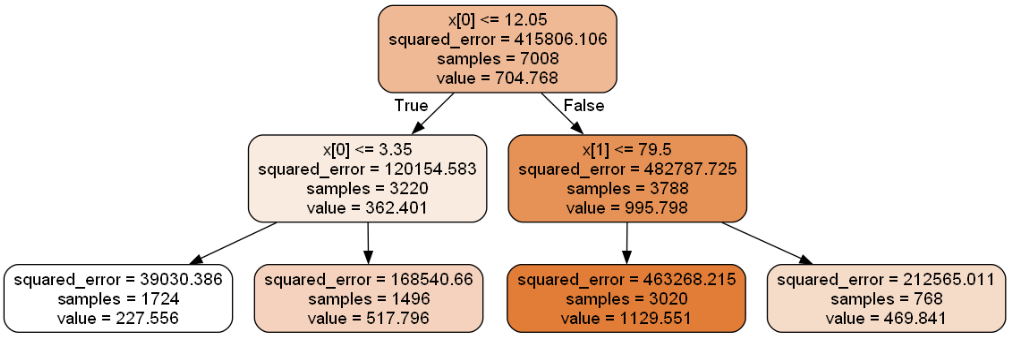
- Cross-validation report:
  `0.2584650669710952 +/- 0.017031035301739363`

Polynomial Kernel:

- Kernel Support Vector Machines with a Polynomial Kernel give an R-squared value of `0.1316282941767571 4`, which is again not a good score.
- The R-squared value has no significant difference even after performing the Standard Scaling.
- Cross-validation report:
  `0.2584650669710952 +/- 0.012863934014534393`

Decision Tree Regressor:

- Decision Tree Regressor with a **`max_depth = 10`** gives an R-squared value of `0.793218887713431`
- This is a better score than the models performed above and a decent score for the regression problem.
- The performance of the model was reduced to `0.7895521941264743` after training the model with standard scaling.
- The decision tree was visualized at a depth of 2 and its visualization became hard after the depth of 2.



- Cross Validation report:
  `0.8070931596269155 +/- 0.021132212689913797`

Random Forest Regressor:

- Random Forest Regressor with the **`max_depth = 14`** gives an R-squared value of `0.8657819527855988`
- This is a pretty good score for the given problem and the features.
- There is no significant difference in the accuracy even after standard scaling.
- Cross Validation report:
  `0.872916798763438 +/- 0.013757294482023605`

Ada Boost Regressor:

- Ada Boost Regressor with the **`n_estimators= 50`** gives an R-squared value of `0.8734656131613165`
- This score is better than that of the Random Forest Regressor and is good for the given problem.
- It resulted in a decreased accuracy after standard scaling with an R-squared value of `0.8698834445098265`.
- Cross Validation report:
  `0.8648465862813873 +/- 0.014426526643885986`

Gradient Boost Regressor:

- Gradient Boost Regressor with the `n_estimators= 100` and `max_depth=7` gives a R-squared value of `0.880793854740713`
- This score is better than all other models and an excellent score for the given problem.
- The model accuracy increased to `0.8888344121656205` after standard scaling.
- Cross Validation report:
  `0.8888344121656205 +/- 0.011202078969374576`

Different Machine Learning Regression Models with their Scores:

Accuracies with Feature Selection:

Although there are better strategies than feature selection for this problem statement, it is used on a few models to compare accuracy.

Feature Selection: Select K Best	Score: R-squared value (Without Feature Selection)	Score: R-squared value (With Feature Selection)
Random Forest Regressor	0.6013121755610251	0.5772841948452658
Linear Regression	0.41054261609446363	0.4056628271111451
Decision Tree Regressor	0.5250814818131624	0.5370755327386377

Feature Selection: Select Percentile	Score: R-squared value (Without Feature Selection)	Score: R-squared value (With Feature Selection)
Random Forest Regressor	0.6013121755610251	0.23884037739300834
Linear Regression	0.41054261609446363	0.3042588236521452
Decision Tree Regressor	0.5250814818131624	0.3452398214404284

Feature Selection: Variance Threshold	Score: R-squared value (Without Feature Selection)	Score: R-squared value (With Feature Selection)
Random Forest Regressor	0.6013121755610251	0.6025875413687579
Linear Regression	0.41054261609446363	0.40997569976831294
Decision Tree Regressor	0.5250814818131624	0.5285379895954558

Accuracies without Feature Selection:

Model	Score: R-squared value (Without Standard Scaling)	Score: R-squared value (With Standard Scaling)
Linear Regression	0.5473867855588497	0.547439352296053
Polynomial Regression	0.5483926871573729	0.5674320849008825
LASSO	0.5474048249724595	0.547477679730227
Ridge Regression	0.5473869488866467	0.5474401909508667
Elastic Net	0.5457862644215118	0.547333580316779
Linear Support Vector Machines	0.10585998679227937	-0.007484114071456993
Kernel Support Vector Machines(RBF)	0.24025324909590007	0.24885016265253423
Kernel Support Vector Machines(Poly)	0.13162829417675714	0.13442038419481528
Decision Tree Regressor	0.793218887713431	0.7895521941264743
Random Forest Regressor	0.8657819527855988	0.8643003641102847
Ada Boost	0.8734656131613165	0.8698834445098265
Gradient Boosting	0.8871648614090362	0.880793854740713

Cross Validation report:
A 10-fold cross-validation is performed on every model.

Model	Mean	Standard Deviation
Linear Regression	0.565695570912433	0.018314632343610502
Polynomial Regression	0.565695570912433	0.018314632343610502
LASSO	0.5656946279604732	0.01827394289560667
Ridge Regression	0.5656956181278698	0.01831411205220463
Elastic Net	0.5656873672830806	0.018158121725113555
Linear Support Vector Machines	0.3390961594703465	0.15704580161109005
Kernel Support Vector Machines(RBF)	0.2584650669710952	0.017031035301739363
Kernel Support Vector Machines(Poly)	0.1348449204990502	0.012863934014534393

Decision Tree Regressor	0.8070931596269155	0.021132212689913797
Random Forest Regressor	0.872916798763438	0.013757294482023605
Ada Boost	0.8648465862813873	0.014426526643885986
Gradient Boosting	0.8888344121656205	0.011202078969374576

5. Discussion

Hyperparameter tuning:

- Hyperparameters are the parameters of the learning process, unlike the parameters of the model.
- Tuning these parameters has a significant effect on the model performance.
- There are different ways to tune these parameters. Grid search and Random search are the two popular methods. The Sklearn library has the functionality to accomplish this.
- The above accuracies are obtained after tuning the hyperparameters. The Grid and Random searches were performed depending on the Machine Learning technique. Manual tuning is also applied for some required models.
  - For regularization methods, the hyperparameter `"alpha"` is searched from the range 0.1 to 1, and the best accuracy was obtained at `alpha =0.1.`
  - Support Vector Machines are computationally intensive to do hyperparameter tuning with Sklearn, and the best hyperparameters are obtained by manual hyperparameter tuning and the best hyperparameters are:
    - Linear SVMs: `epsilon = 1,C=100`
    - Kernel RBF: `epsilon = 0.1, gamma = 0.01, C = 50`
    - Kernel Poly: `epsilon = 0.1, C = 1000, degree = 5`
  - For decision trees, `max_depth: 10` is the best value for the maximum height of a tree.
  - Random forest regressor has two parameters, and the best values are `n_estimators=30,max_depth=14`
  - Ada boost Regressor gives maximum accuracy at `n_estimators = 10`
  - Gradient Boost Regressor is the best-performing model at the `n_estimators= 100, max_depth=7`

Comparing results:

- Techniques such as the Ada boost Regressor and the Gradient Boost regressor yield a much better score than all other models.
- Gradient Boost Regressor is the robust technique for this problem statement. Its accuracy is 89%, which is close to 90%.
- The initial baseline linear regression model without using techniques like Ordinal Encoding and Power Transformation had an accuracy of 40%, but now, with these robust techniques and models, the accuracy has reached close to 90%.
- 90% accuracy is acceptable for this particular regression problem.
- All the models were evaluated using the cross-validation technique. All the models excelled in 10 cross-validation tests with a good mean and low standard deviation except the Support Vector Machines. Failure in the cross-validation test is acceptable since SVMs did not perform well.
- From this, we can conclude that the trained models give good results and are not sensitive to new instances.