Seoul Bike Sharing Demand Prediction

Introduction

In this project, we will be implementing linear and logistic regression on Seoul Bike sharing Demand Data to predict rented bike demand. We will implement the gradient descent algorithm with batch update. In addition, we will experiment with design and feature choices.

The dataset source: https://archive.ics.uci.edu/ml/datasets/Seoul+Bike+Sharing+Demand

Attributes

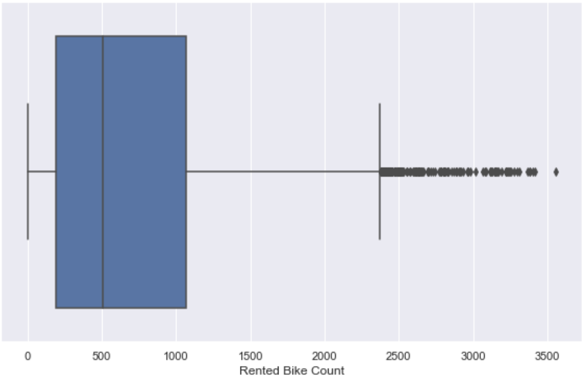
Date: year-month-day
Rented Bike count - Count of bikes rented at each hour
Hour - Hour of the day
Temperature-Temperature in Celsius
Humidity - %
Windspeed - m/s
Visibility - 10m
Dew point temperature - Celsius
Solar radiation - MJ/m2
Rainfall - mm
Snowfall - cm
Seasons - Winter, Spring, Summer, Autumn
Holiday - Holiday/No holiday
Functional Day - NoFunc(Non Functional Hours), Fun(Functional hours)

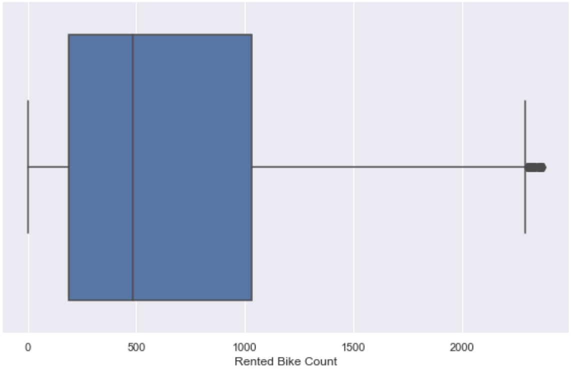
Data Pre-Processing

1. Check for Null/Missing values: There are no missing values in the dataset.

2.Feature creation: Created 2 variables which are month and Weekday. The values of the 2 variables are extracted from Date variable. Month variables has values from 1-12 indicating months from January-December. Weekday variable has values from 0-6 indicating the day of week with Monday=0 and Sunday=6.

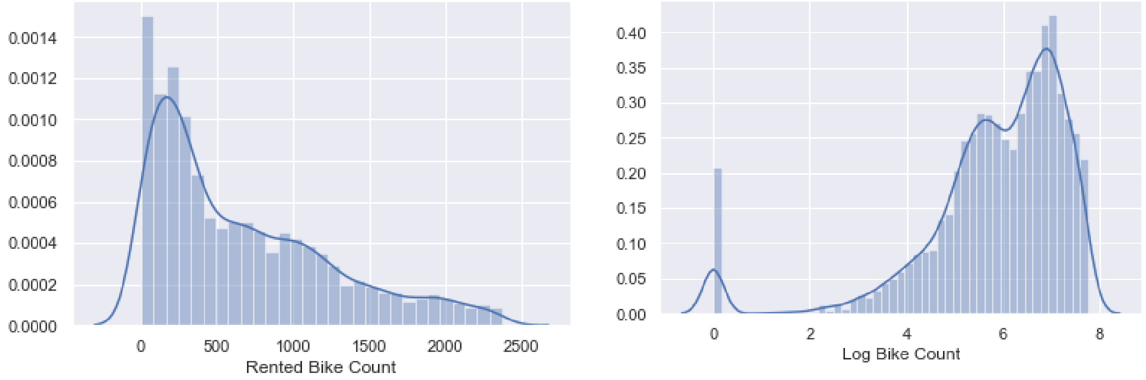
3.Target variable outlier detection: From the box plot we can see that Rented Bike Count has a lot of outliers. The variable range is (0,3556) with 3rd quartile being 1065.25. The values greater than 1.5*IQR (Interquartile Range) from 3rd quartile are removed.



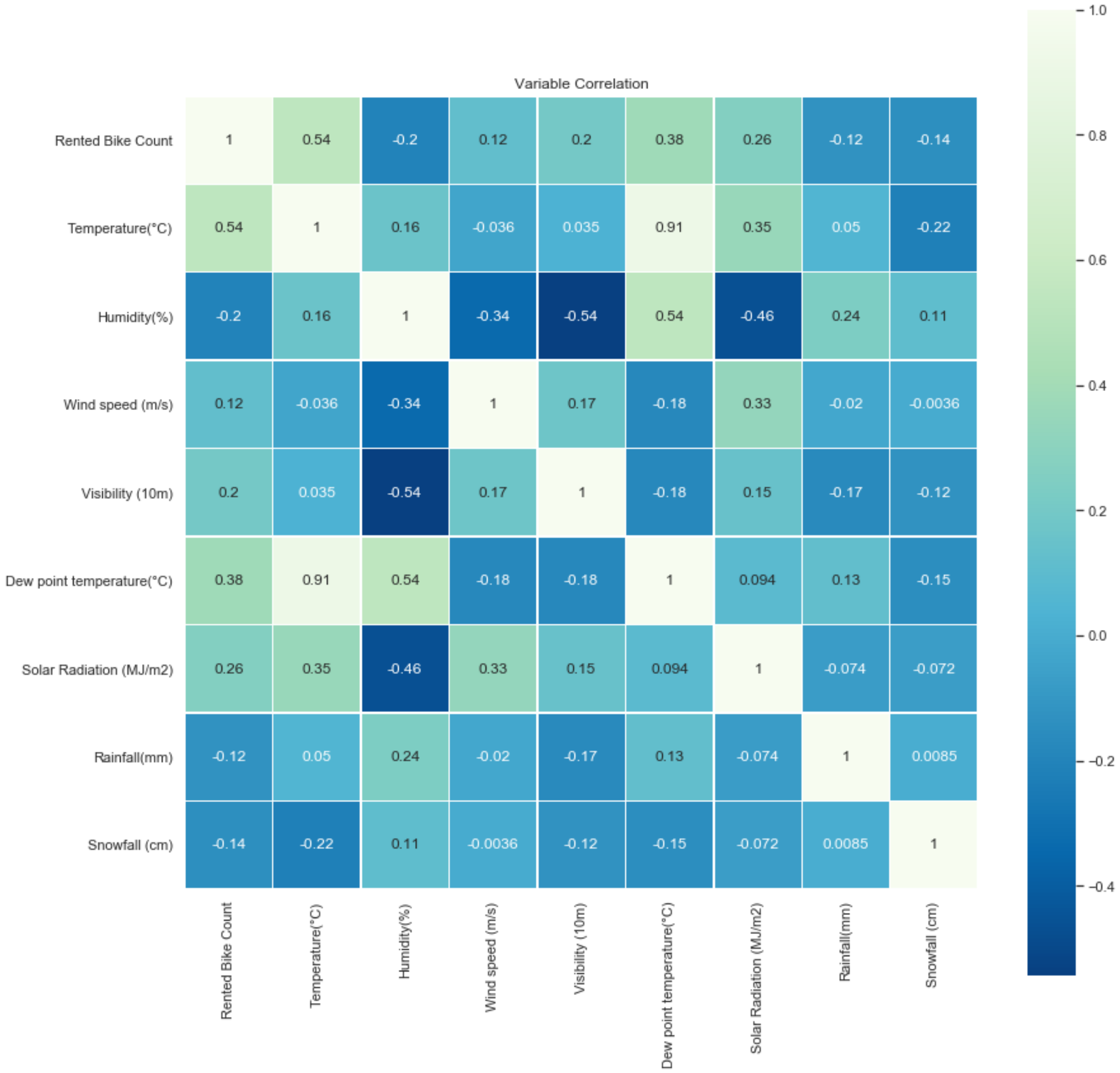


4.Dummy variable creation: Dummy variables are created for categorical variables Seasons, Holiday, Functioning Day, Month and Weekday.

5. Target variable distribution: Rented Bike Count has right skewed distribution. The variable is transformed using log transformation to make the distribution approximately normal.



6.Check for Multicollinearity: From the heatmap shown below, there is no significant correlation between the numerical variables. Rented Bike Count has the highest correlation of 0.54 with Temperature.



7.Feature scaling: Numerical variables are standardized using formula: $z = \frac{x-u}{\sigma}$

8.Intercept: Added intercept columns with value 1 for intercept term of the equation.

Finally, the dataset is split into training set and set test in 80:20 ratio.

Algorithms

Linear Regression

The linear regression algorithm has 5 user defined functions which are:

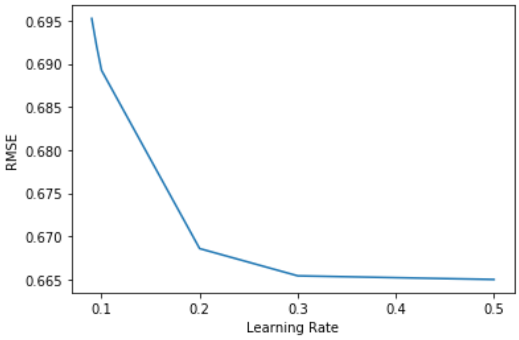
- linear_cost_func: Returns the Mean square error cost function value.
- linear_gdesc: Returns the coefficient matrix for the minimum cost after performing gradient descent.
- predict: Returns list of predicted values.
- Reg_rmse: Returns RMSE (residual mean square error).
- LinearReg: Performs linear regression calculations using gradient descent on the dataset passed into the function. Returns RMSE.

Parameter tuning for Linear Regression:

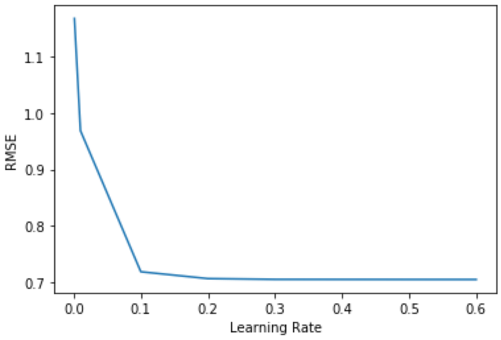
Experimenting with various values of learning rate (α) to find the optimal value of α which gives the least RMSE.

With threshold of 0.00001, α of 0.3 gives the least RMSE with both test and train dataset.

Test Set RMSE vs Learning Rate

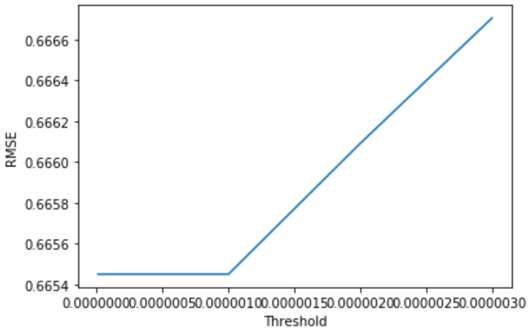


Train Set RMSE vs Learning Rate

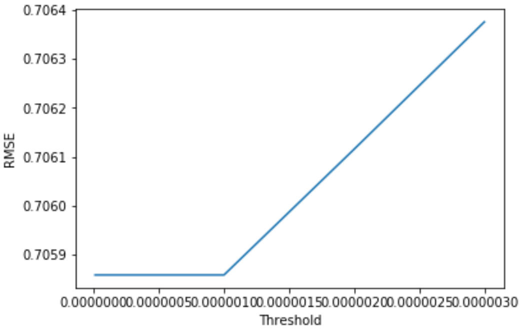


Experimenting with various values of threshold to find the optimal value which gives the least RMSE. With α of 0.3, threshold value of 0.000001 gives the least RMSE with both test and train dataset.

Test Set RMSE vs Threshold

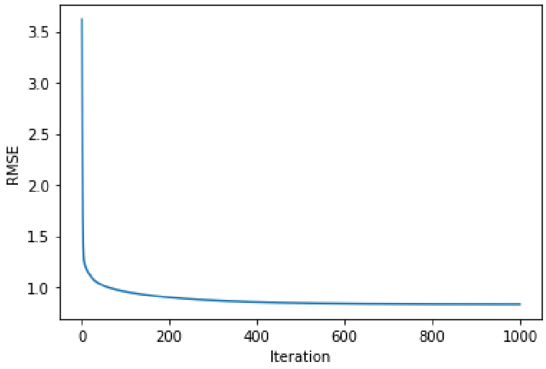


Train Set RMSE vs Threshold

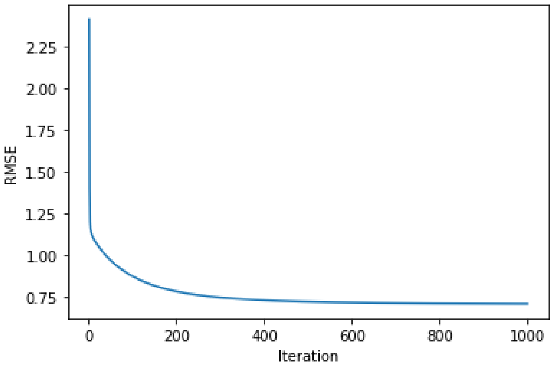


Below are the plots for RMSE as a function of number of iterations for test and train set with α=0.3 and threshold = 0.00001. We can see that increasing the number of iterations beyond 600 does not improve the RMSE. The Lowest RMSE obtained is 0.66 on test data set and 0.70 on train data set.

Test Set



Train Set



Feature Selection for Linear Regression:

Selected 8 random features to retrain the model, at optimal value of learning rate and threshold. . Comparing the results to model with all features.

Linear Regression	Test Set	Train Set
Randomly selected features:	Humidity(%), Weekday_5, Functioning_Yes, Season_Summer, Weekday_3, Weekday_1, Month_10, Season_Winter	Humidity(%), Weekday_5, Functioning_Yes, Season_Summer, Weekday_3, Weekday_1, Month_10, Season_Winter
α	0.3	0.3
Threshold	0.000001	0.000001
RMSE 8 features	0.83	0.86
RMSE all features	0.66	0.70

The RMSE increases when we use 8 features compared to all features. This can indicate that we are not using relevant features. The features selected randomly are not explaining any variation in data.

Selected 8 relevant features to retrain the model, at optimal value of learning rate and threshold. Comparing the results to model with all features.

Linear Regression	Test Set
Most relevant features:	Season_Spring, Rainfall(mm), Hour,Dew point temperature(°C), Solar Radiation (MJ/m2), Weekday_5,Humidity(%), Functioning_Yes
α	0.3
Threshold	0.000001
RMSE 8 relevant features	0.73
RMSE all features	0.66

RMSE decreases when 8 relevant features are used for regression compared to randomly selected features.

Logistic Regression

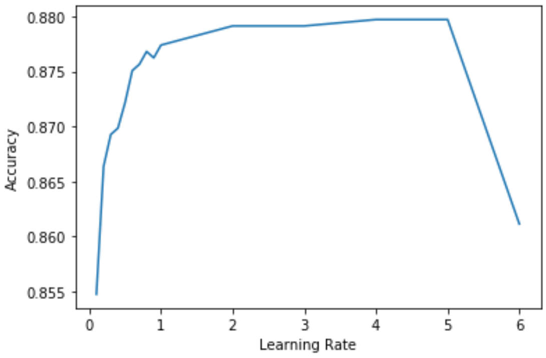
The target variable is converted into binary variable using mean as cut off value. Values greater than mean are assigned 1 else 0. Logistic regression algorithm has 6 user defined functions which are:

- sigmoid: Returns the sigmoid function value.
- logit_cost_func : Returns cost function value.
- logit_gdesc: Returns the coefficient matrix for the minimum cost after performing gradient descent.
- log_predict: Returns list of predicted values.
- accuracy: Returns accuracy.
- LogisticReg: Performs logistic regression calculations using gradient descent on the dataset passed into the function. Returns accuracy.

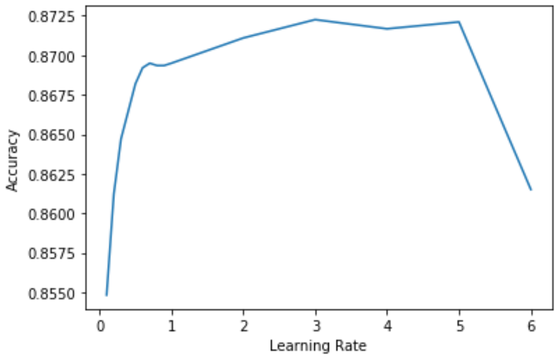
Parameter tuning for Logistic Regression:

Experimenting with various values of learning rate (α) to find the optimal value of α which gives the maximum accuracy. With threshold of 0.00001, α of 5 gives the maximum accuracy with both test and train dataset.

Test Set Accuracy vs Learning Rate

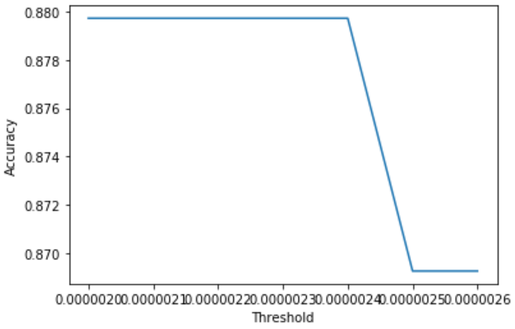


Train Set Accuracy vs Learning Rate

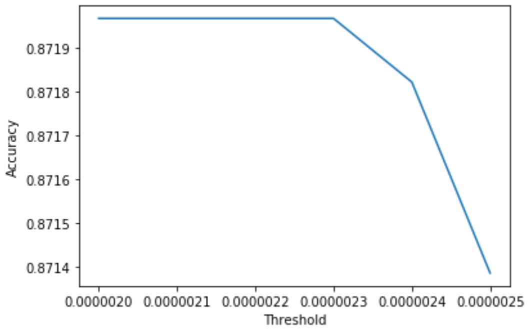


Experimenting with various values of threshold to find the optimal value which gives the maximum accuracy. With α of 5, threshold value of 0.0000024 gives the maximum accuracy with test set and threshold value of 0.0000023 gives maximum accuracy with train dataset.

Test Set accuracy vs Threshold

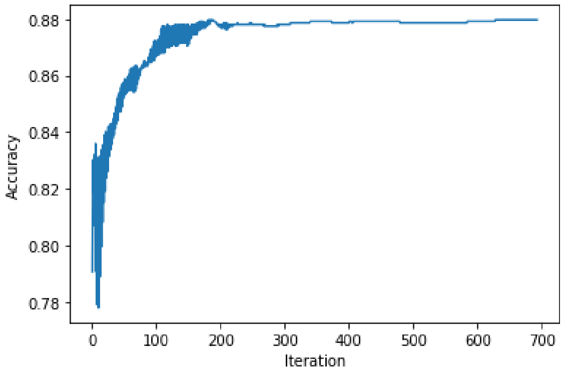


Train Set accuracy vs Threshold

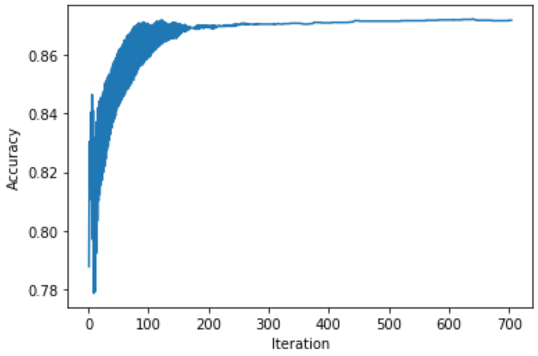


Below are the plots for accuracy as a function of number of iterations for test and train set with α=5 and threshold = 0.000024 for test set and 0.0000023 for train set. We can see that increasing the number of iterations increases the accuracy but threshold is reached around 700 iterations. The maximum accuracy obtained is 0.88 on test data set and 0.87 on train data set.

Test Set



Train Set



Feature Selection for Logistic Regression:

Selected 8 random features to retrain the model, at optimal value of learning rate and threshold. . Comparing the results to model with all features.

Logistic Regression	Test Set	Train Set
Randomly selected features:	Humidity(%), Weekday_5, Functioning_Yes, Season_Summer, Weekday_3, Weekday_1, Month_10, Season_Winter	Humidity(%), Weekday_5, Functioning_Yes, Season_Summer, Weekday_3, Weekday_1, Month_10, Season_Winter
α	5	5
Threshold	0.0000024	0.0000023
Accuracy 8 features	0.76	0.76
Accuracy all features	0.88	0.87

The accuracy reduces when we use 8 features compared to all features. This can indicate that we are not using relevant features. The features selected randomly are not explaining any variation in data.

Selected 8 relevant features to retrain the model, at optimal value of learning rate and threshold. Comparing the results to model with all features.

Logistic Regression	Test Set
Most relevant features:	Season_Spring, Rainfall(mm), Hour,Dew point temperature(°C), Solar Radiation (MJ/m2), Weekday_5,Humidity(%), Functioning_Yes
α	5
Threshold	0.000024
Accuracy 8 relevant features	0.84
Accuracy all features	0.88

Accuracy is higher than randomly selected features model because we are selecting more significant features. But highest accuracy is obtained by using all features for logistic regression.

Conclusion:

To predict if Rented bike demand would be high or low, we can use the logistic regression model.

To predict the approximate count of Rented bike demand, the best model to be used would be linear regression model with α=0.3 and threshold = 0.000001. The best 8 significant features are Season_Spring, Rainfall(mm), Hour, Dew point temperature(°C), Solar Radiation (MJ/m2), Weekday_5, Humidity(%) and Functioning_Yes.

Rented Bike Count =-0.051+ 0.056*Season_Spring-0.234*Rainfall(mm)+0.263*Hour+0.776*Dew point temperature(°C)-0.024*Solar Radiation (MJ/m2) +0.018*Weekday_5-0.623*Humidity (%) +6.096*Functioning_Yes.

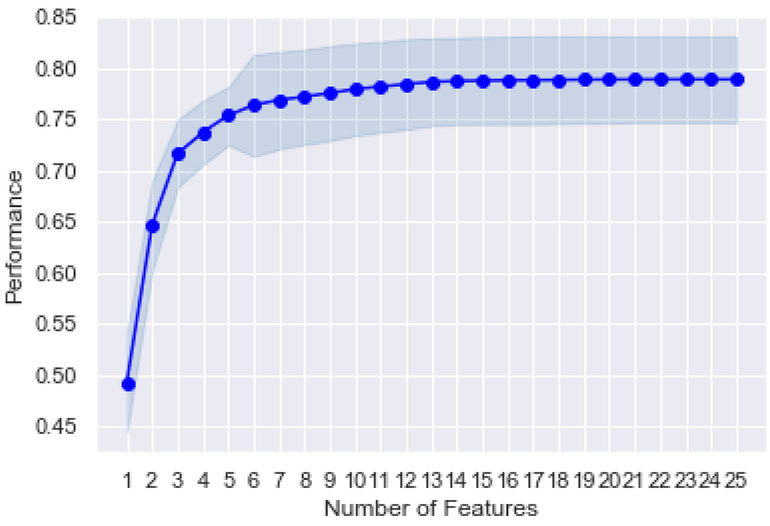
Interpretation of features:

- When season is spring the Rented bike count increase by a factor of $e^{0.056}$=1.07 or by 7% when season is spring compared to when it is Autumn.
- Rented bike count increases by 23% when rainfall increases by 1mm.
- Rented bike count increases by 78% for 1°C increase in temperature.
- Rented bike count decreases by 62% for 1% increase in humidity.
- When day is Saturday the Rented bike count increases by 2% when day is Monday.

Model fit improves as significant features are added. We can reduce the RMSE further by increasing the number of significant features to be added.

Subset selection:

Performed stepwise subset selection to find the best number of significant features.



Best 18 features from Stepwise Selection (k=18):
'Hour', 'Temperature(°C)', 'Humidity (%)', 'Dew point temperature(°C)', 'Rainfall(mm)', 'Season_Spring', 'Season_Winter', 'Holiday_No Holiday', 'Functioning_Yes', 'Month_5', 'Month_7', 'Month_8', 'Month_9', 'Month_12', 'Weekday_1', 'Weekday_2', 'Weekday_4', 'Weekday_6'

R square value of best model:
0.8038084371015503