APPLIED MACHINE LEARNING

ASSIGNMENT 1

Srinikethan Pusthay (SXP210162)

Goal: Implement a linear regression model on the dataset to predict the rented bike count.

Dataset:
The dataset contains count of public bikes rented at each hour in Seoul Bike haring System with the corresponding Weather data and Holidays information.

Tasks:

Part1:
The dataset is downloaded from the source in csv format and is split into train and test by 70% and 30%. Before splitting we have modified the data. Some features have data in categories. These have been converted into factors.
We have four values for seasons in Seasons column, Spring, Summer, Autumn, and winter. These have been modified as 1,2,3,4 respectively. Functioning day has two inputs yes / no. It has been modified as 1 for yes and 0 for no. And similarly in holiday column, 0 for No Holiday and 1 for Holiday.

Part2:
Linear regression model:

(Rented_bike_count) = b0 + (b1 * hour) + (b2* temperature_in_c) + (b3 * humidity_percentage) + (b4 * wind_speed_mps) + (b5 * visibility_10m) + (b6 * dew_point_temperature_in_c) + (b7 * solar_radiation) + (b8 * rainfall_mm) + (b9 * snowfall_cm) + (b10 * seasons) + (b11 * holiday) + (b12 * functioning_day)

We have implemented the linear regression model using gradient descent algorithm with batch update. Rented Bike Count is the Y variable which is obtained from the features. Features are written along with the parameters from b0 to b12.

Part3:
Using learning rate as 0.01 and 1000 iterations as initial values, we have obtained the initial parameter values as below:

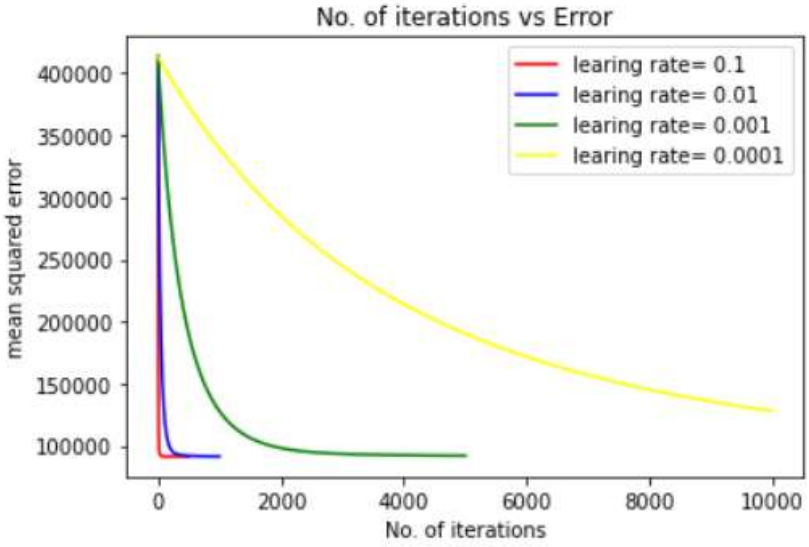
(Rented_bike_count) = 649.875 + (175.834 * hour) + (242.256* temperature_in_c) - (167.913 * humidity_percentage) + (26.755 * wind_speed_mps) + (34.658 * visibility_10m) + (146.647 * dew_point_temperature_in_c) - (44.115 * solar_radiation) - (64.358 * rainfall_mm) + (19.111 * snowfall_cm) - (79.206 * seasons) - (19.913 * holiday) + (70.609 * functioning_day)

Experiments and Results:

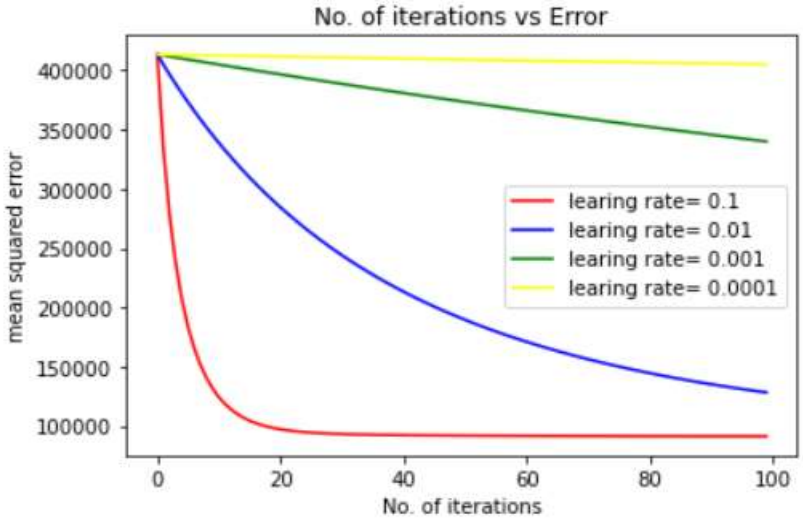
Experiment1: Experiment with learning rate and iterations for linear regression

We have performed various experiments by changing the values of learning rate and iterations. Please find its respective results in graphs below.

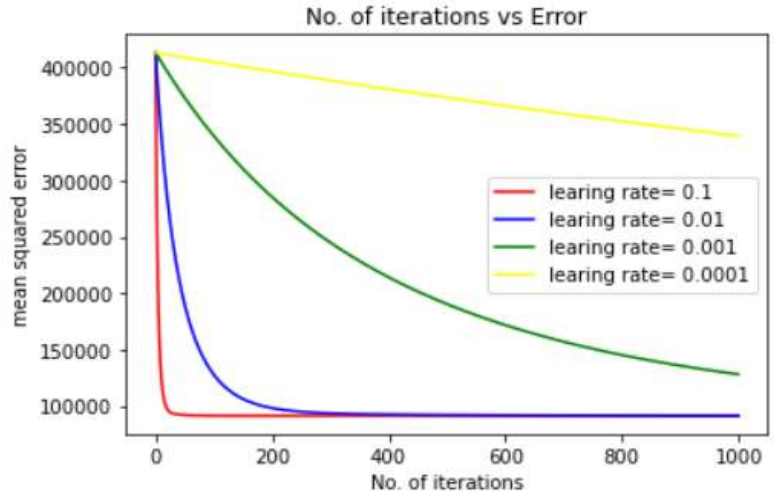
- Different learning rates at different iterations
  From the picture below we can see the curves behavior with respect to iterations. For smaller learning rate, the curve is taking many iterations to converge. For higher learning rate, it is taking only fewer iterations to converge and reach the minimum mean squared error. We shall next observe how learning rate behaves with different iterations.



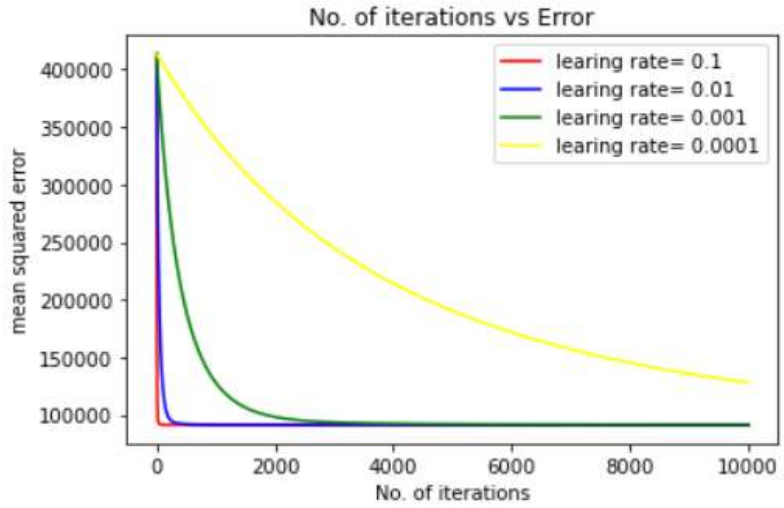
- Different learning rates at 100 iterations



- Different learning rates at 1000 iterations

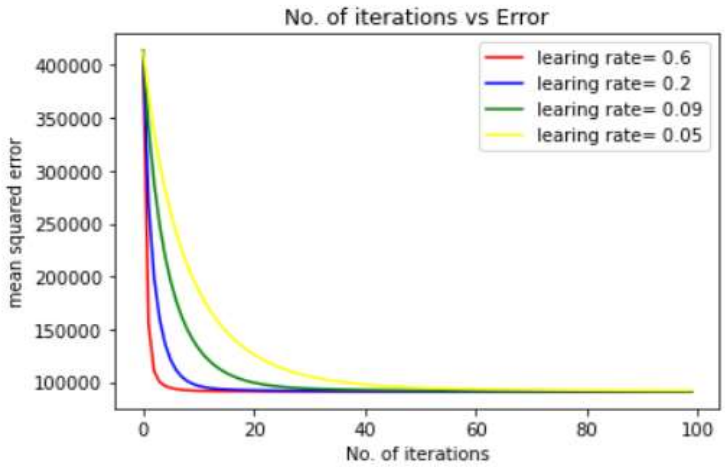


- Different learning rates at 10000 iterations

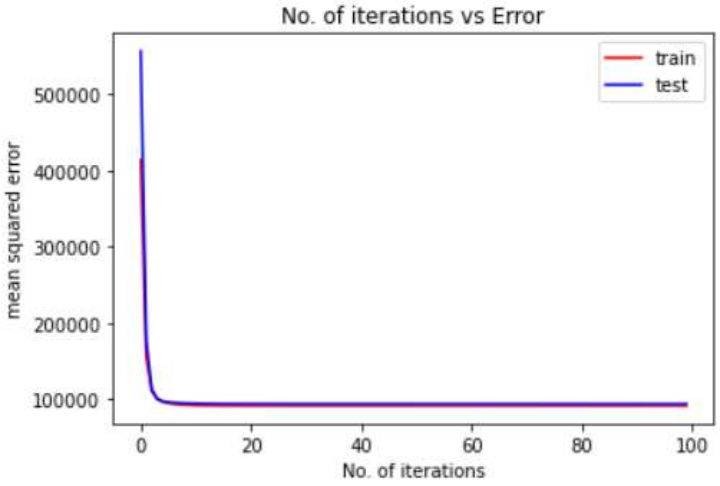


From the above graphs, the error has been continuously decreasing and the convergence have reached when there is not much change in the graph and is almost a straight line. When the learning rate is low, algorithm takes large number of iterations to converge and when learning rate is very high, the algorithm diverges, and error might not reach the minimum.

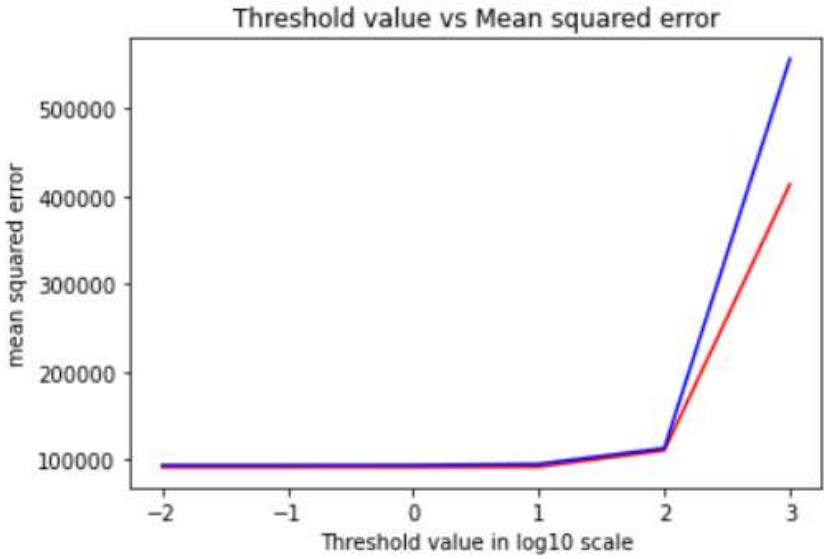
Based on further experiments, we have arrived at the optimum learning rate as 0.6 for 100 iterations. We can observe the minimum error value at 0.6.



Below is the graph for train and test data. Error value starts higher for test data than train data, but minimum error is almost the same for both.

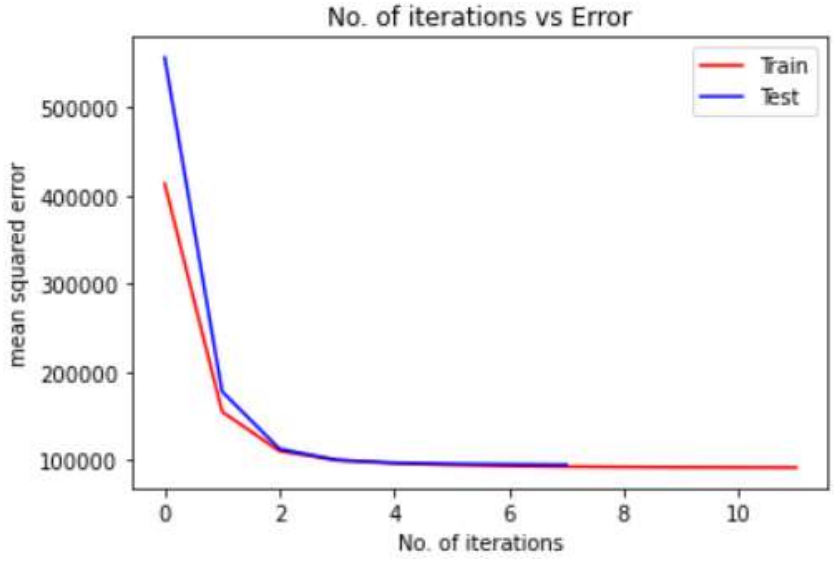


Experiment2: Experiment with various thresholds for convergence for linear regression



The minimum test error is found at threshold of 10. When the change in error is within the convergence threshold, the algorithm is said to be converged. So, we found the optimum value of threshold as a function of threshold and minimum error.

Error results of train and test set at optimum threshold value.

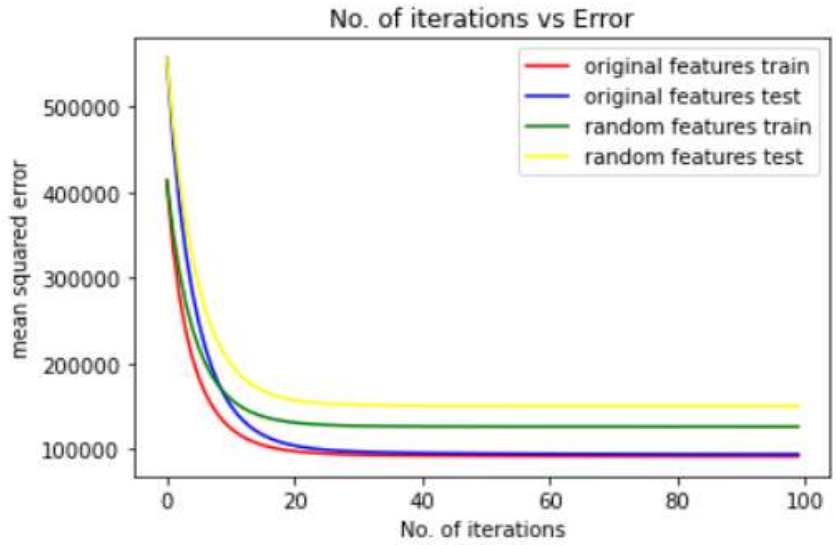


Experiment3: Experiment with eight random selected features

Features have been selected randomly and below is the equation.

(Rented_bike_count) = b0 + (b1 * visibility_10m) + (b2 * dew_point_temperature_in_c) + (b3 * solar_radiation) + (b4 * rainfall_mm) + (b5 * snowfall_cm) + (b6 * seasons) + (b7 * holiday) + (b8 * functioning_day)

Below graph, provides comparison of the original features and randomly selected features. We can observe that the model of randomly selected features is poor compared to original selected features.



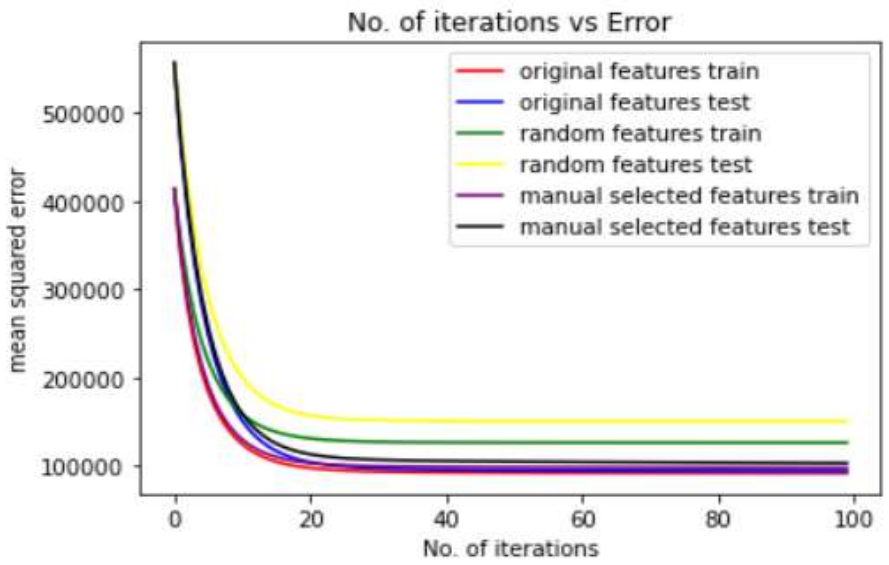
Experiment4: Experiment with eight best features selected manually

Based on the data, I have picked the following eight features to be the best fit for the equation to predict the output.

(Rented_bike_count) = b0 + (b1 * hour) + (b2* temperature_in_c) + (b3 * wind_speed_mps) + (b4 * visibility_10m) + (b5 * dew_point_temperature_in_c) + (b6 * solar_radiation) + (b7 * rainfall_mm) + (b8 * functioning_day)

These features were selected as I felt they are more important from the remaining. Temperature check is a must for any customer, and similarly solar radiation if it is summer and rainfall if it is rainy season. And wind speed also depends on the climate which is an important factor for a customer to rent a bike. Another factor is the functioning day.

We can observe from the graph the error of manual selected features is between random features and original features.




```
Original Set of features: Train data maximum and minimum error
413352.87092302676 91375.1521584842
Original Set of features: Test data maximum and minimum error
556240.4644216134 93732.8699927596
Random Set of features: Train data maximum and minimum error
413352.87092302676 125923.49300360223
Random Set of features: Test data maximum and minimum error
556240.4644216134 150229.5770076141
Manually selected Set of features: Train data maximum and minimum error
413352.87092302676 97258.7099154246
Manually selected Set of features: Test data maximum and minimum error
556240.4644216134 102808.01089571702
```

There is a change in errors between the set of features provided above. Manually selected features performed better than randomly selected features but as we are missing few of the other features, it is not completely better model than original set of features.

Through experimentation, we can say parameters like learning rate, threshold etc. provides better performing model to predict the outputs. We can also say how these factors are affected with other dependent factors. For reaching to optimum learning rate, we need to try experimenting with iterations as learning rate affects the convergence of the error.
We can also say features selection is important in the performance of the model. With better domain knowledge and idea of the features can help more in improving the model.