Prepared by: Jyotsna Nagaprasad Ambekar Date: 06/10/2022

BUAN 6341 Applied Machine Learning

ASSIGNMENT NO 1

Seoul Bike Data

Executive Summary

- **Tuning Learning rate, Maximum epochs, Stopping Threshold improved the model by decreasing the error values and best estimation of target variable through features.**
- **Feature selection with proper domain knowledge helps in improving model performance.**
- **Temperature and Hour are best predictors in linear model.**
- **Prediction accuracy can be further improved by implementing variable transformations, introducing interaction terms, assessing outliers.**

Context

Currently Rental bikes are introduced in many urban cities for the enhancement of mobility comfort. The goal is to help commuting in the cosmopolitan cities and reduce the pollution amounts caused by cars, individual motorcycles etc. and create green cities altogether. It also is a healthiest way for travelling to work or school. It is important to make rental bike available and accessible to the public at the right time lessening the waiting time. Eventually, providing the city with stable supply of rental bikes becomes a major concern as it can be used as alternative to the commuters in cities. The crucial part is the prediction of bike count required at each hour for the stable supply of rental bikes.

Introduction

In this project, the objectives were to implement a gradient descent algorithm with batch update to predict the Rented Bikes count on a given day with given attributes. This report details various experiments conducted using the dataset to understand the effect of tuning hyperparameters of the gradient descent algorithm. Also, the effect of the use of various weather conditions as independent variables/ features in prediction is evaluated and the best model was selected through experimentation.

About the Data

The dataset consists of 14 features and 8760 records with no missing values. The data contains information regarding weather and how much solar radiation is observed on the day, seasons, holiday, functional hours etc. The target variable is the rented bike count per hour.

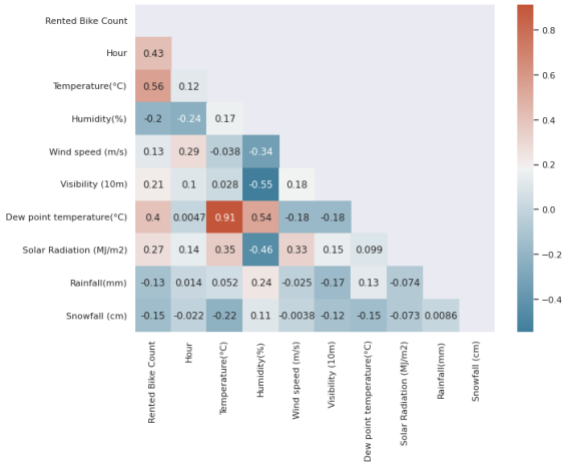
Attribute information

Date : year-month-day
Rented Bike count - Count of bikes rented at each hour
Hour - Hour of the day
Temperature-Temperature in Celsius
Humidity - %
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)

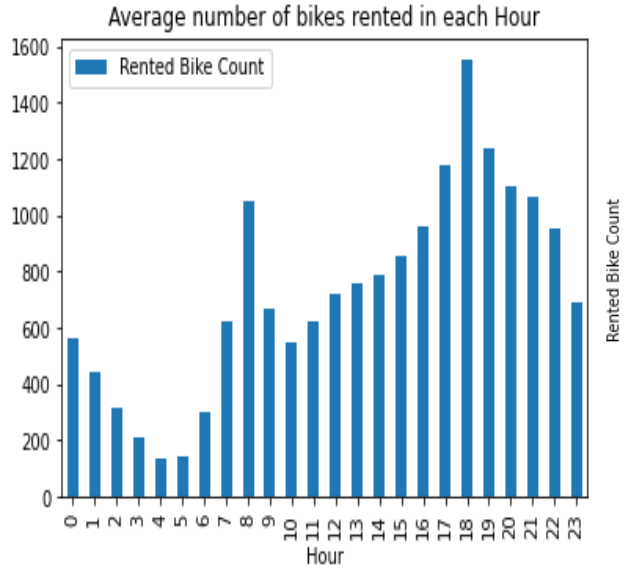
Data Preprocessing & Visualization

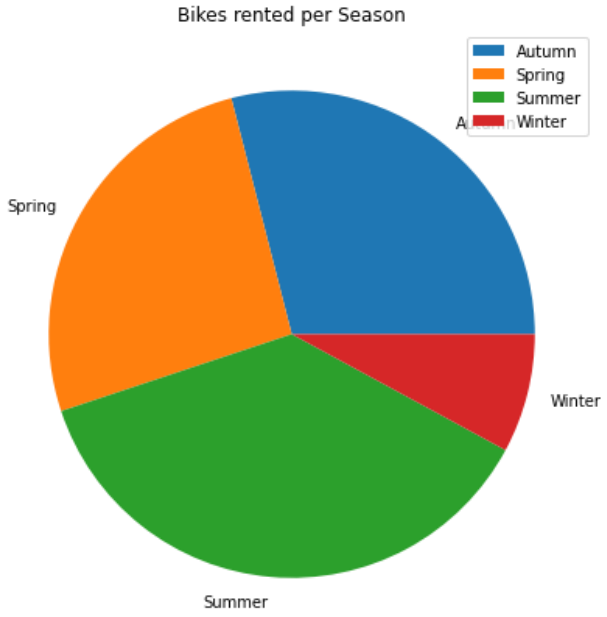
It is the technique in which data is processed so that it is appropriate to feed the model. Preprocessing steps applied were One hot encoding to encode categorical variables like Seasons & Holiday columns, Splitting the dataset into training and test sets, Feature scaling for standardizing data. Firstly, data exploration was done by understanding the types of data present and description of data. The Date column which was initially string was converted to DateTime format.

The temperature and dew point temperature have correlation value of 0.91. Due to this high correlation, multi collinearity arises if both the features are used to develop the model. Hence dew point temperature is dropped from the independent variables.



The average number of bikes rented each hour is shown as a bar plot.





The pie chart shows the bikes rented during each season. In summer, the bikes rented is maximum compared to other seasons.

The number of bikes rented is maximum on Friday with total number of 950334. Also, 54.35% more bikes are rented in the Day time compared to night times. The same trend is observed on Non-Holidays than on Holidays. The whole data is split into Train & test datasets in 70-30 ratio. The sets are separately scaled so that the information about test set is not leaked, and we have a fair understanding about the test data when we evaluate metrics.

Algorithm Implementation

The gradient descent algorithm is implemented along with cost function using python for the Linear Regression modelling technique. Gradient descent is an optimization algorithm used to find the values of coefficients of features that minimizes cost function. The algorithm starts with assuming random weights and evaluating cost function. The cost is calculated over the entire training set for each iteration in the batch gradient descent algorithm. The learning rate controls how much coefficients can change in each update. The maximum epochs show maximum number of times this update is allowed. The hyperparameters tuned during the process are Learning Rate, Stopping Threshold, Epochs.

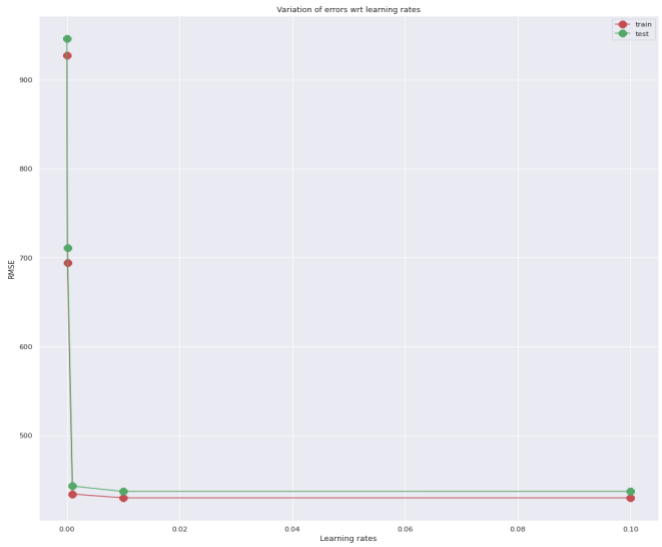
The independent variables used for training the model are Hour, Temperature(°C), Humidity(%), Wind speed(m/s), Visibility (10m), Solar radiation (MJ/m2), Rainfall(mm), Snowfall, Seasons & Holidays one hot encoded columns. The season column encoded as 0 is Autumn, 1 is spring, 2 is summer, 3 is winter. The Holiday column encoded as 4 is Holiday and 5 is No-Holiday

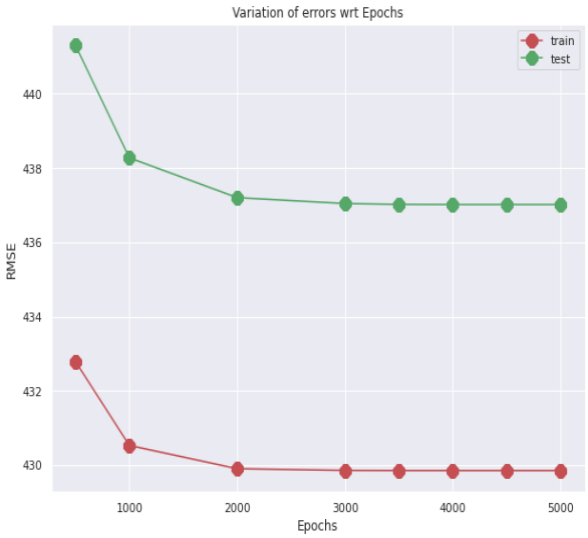
After application of gradient descent algorithm, the linear regression equation turned out to be –

Rented Bike count = 720.5622 + 202.1251 * Hour + 318.3954 * Temperature -166.3857 * Humidity + 17.4747 * Wind speed + 1.6141 * Visibility -71.6971 * Solar radiation -72.4992 * Rainfall + 10.1981 * Snowfall + 74.9143 * SeasonEncoded0 + 9.1361 * SeasonEncoded1 + 3.8936 * SeasonEncoded2 -87.8936 * SeasonEncoded3 -12.8569 * HolidayEncoded4 + 13.7624 * HolidayEncoded5

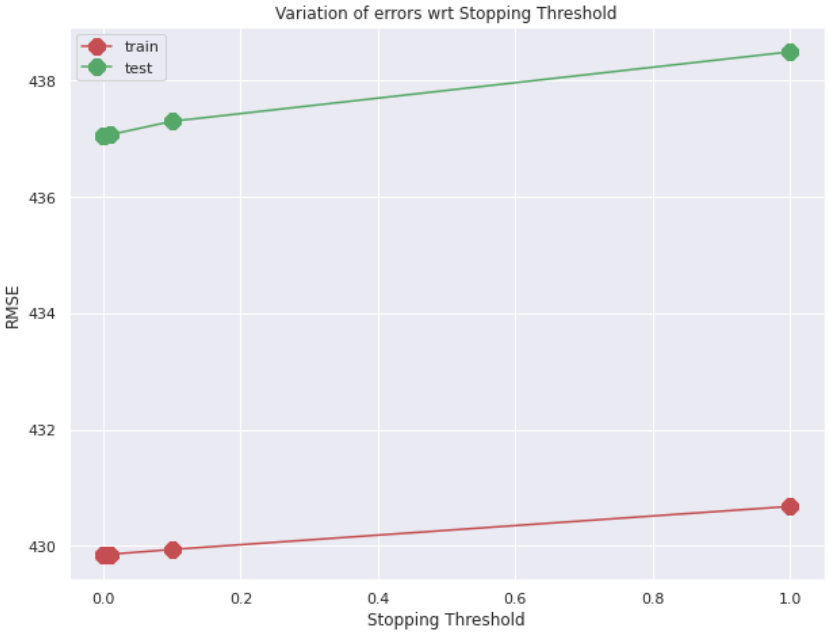
Experimentation

Optimizing hyper parameters help us in minimizing the error values and the model will be able to explain the variation in the target variable better. Firstly various learning rates were given as an input and the RMSE values were plotted against various learning rates. When the learning rate is very small, the RMSE values seem to be very high. There is an exponential decrease in RMSE values as the learning rate reaches ideal one. The ideal learning rate gives lower error values for both train & test sets. The same is described by the graph.





When the variation of RMSE values is observed with respect to epochs, the best value of maximum epoch was found out to be 4000 iterations. Lower the stopping threshold, lower are the RMSE values. With decreasing stopping threshold after an ideal threshold the RMSE value remains same because the algorithm has already converged to the global minimum and increasing epochs or learning rate has no effect. This ideal threshold is found out to be 0.001. As the threshold increases the data is mis fit and the error values increase. The same trend is observed in both training and test sets.

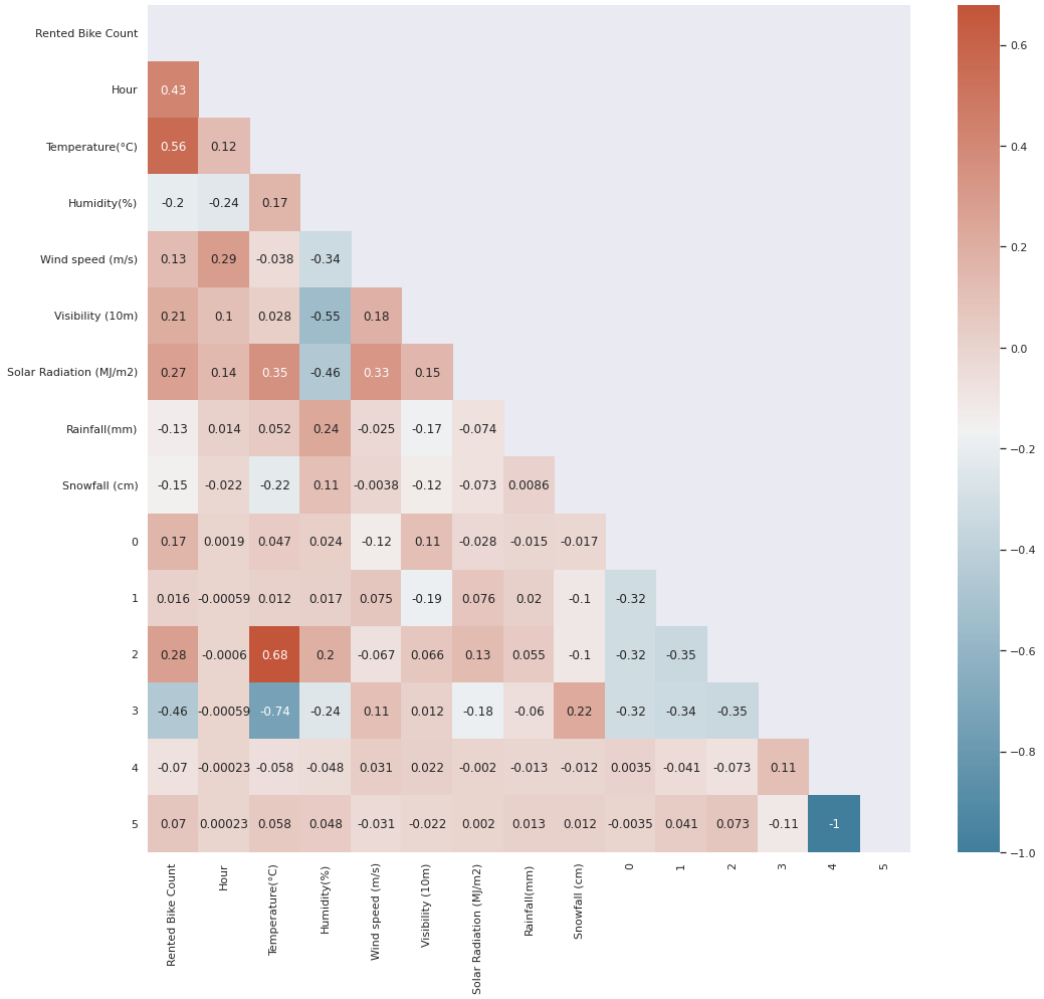


After experimentation, best stopping threshold was 0.001, max epoch value to be 4000, learning rate was 0.01. This was decided based on the Root mean square values & R2 score. The original parameters were calculated based on these values only.

Experimentation was done to observe whether the features selected to train the model has effect on the model performance. Hence the random as well as manual selection of independent variables was done. The manual selection was based on correlation matrix and the RMSE values were observed.

Random Feature Selection: Sampling is done to select independent features at random. Scaling of variables is done separately so that the information about test set is not revealed. The RMSE values are evaluated. The features selected are – ‘SeasonEncoded1’, ‘SeasonEncoded3’, ‘Snowfall’, ‘Visibility’, ‘SeasonEncoded0’, ‘SeasonEncoded2’, ‘Solar Radiation(MJ/m2)’, ‘Rainfall (mm)’

Manual Feature Selection: The correlation matrix shows the correlation between the independent variables and the target variable. The variables with highest correlation values are taken to implement the gradient descent algorithm. Based on correlation matrix the variables selected – 'Temperature(°C)', ' SeasonEncoded3, 'Hour', ' SeasonEncoded2', 'Solar Radiation (MJ/m2)', 'Visibility (10m)', 'Humidity (%)', ' SeasonEncoded0'



The RMSE values are compared

RMSE Values	Best hyper parameters	Random selection of features	Intuitive selection of features
Train	433.0825	555.4754	453.3875
Test	439.7108	566.3962	455.5727

There is 28.26% increase in RMSE value when we compared original features with best hyperparameters and the model with randomly picked features for the training set whereas in test set, the increase is 28.81%

The comparison between original features and manually selected 8 features is done and there is 4.68% and 3.61% increase in errors in training and test sets respectively.

Observations

1. The ideal learning rate was found to be 0.01. Cost function undergoes iterations upto 3800 when a stopping threshold was 0.001 is input into the gradient descent algorithm. All this is done with a goal to minimize Root Mean square values and increasing R2 score there by achieving best possible model performance.

2. The cost function is said to be converged when the percentage change almost reaches 0. This happens faster with the ideal learning rate which is 0.01 for this model.

3. When the learning rate is 1 (high) the cost function doesn't converge.

4. The R2 score for test set was observed to be 0.5477 i.e., 54.77% of variation in target variable is explained by the independent variable.

5. The model with all features and best hyper parameters performs best compared to the other 2 models where the number of features were either randomly selected or manually.

6. The random model performs poorly of all because the random selection doesn't incorporate domain knowledge of the data given. Using the domain knowledge helps us in selecting the features and hence most important for the development of the model to predict new data points.

7. The manual selection performs better than random model because the correlation between the variables was taken into consideration and the highest correlated features were used for the development of the model. But it still performs poorly when compared to the model with all the features because of the limitation of the data available to develop the model.

Temperature and Hour variables seem to be the best predictors for the model. Temperature should be comfortable for someone who is commuting through bikes which is logically true. It is difficult for someone to bike when it is very sunny days in summer or chilly days in winter. Also, the hour of the day is very important factor for renting the bikes as most of the commuting population have a requirement to rent bike at the hours when they have to reach their office or school in the morning or to reach home in the evening. During the peak hours, the need for renting bikes is very high compared to other times.

As a part of next steps, we can do some transformations on variables, include interaction terms in the model for its better performance. The outliers can be treated with acquiring better domain knowledge which also improves model predictability on new data points. We can also consider forward and backward selection process where statistical significance of addition of variables is considered through p-values.

We can use different algorithms to see which explain variation better in the target variable better. Linear regression could explain only 55%. Other models can explain better. The limitations of linear regression namely it being linear in nature can’t explain the outcome variable that efficiently. Models like Kernel SVM can be implemented on the same data to understand the data more explicably and for better model performance.