**PART 1:** Partitioning the dataset into train / test sets using a 70/30 split.

**Code:**

dat = read.csv("**E:/AML - BUAN 6341/hour.csv**", header = TRUE)

train\_data = dat[1:round(0.7\*nrow(dat)),]

test\_data = dat[(round(0.7\*nrow(dat))+1):nrow(dat),]

**PART 2:** Designing a linear regression model using the built-in package (**lm**) to model the count of bike rentals

**Code:**

model <- lm(cnt~yr + holiday+ workingday+ temp+ atemp+ hum+ windspeed+ season\_1+ season\_2+ season\_3+ season\_4+ mnth\_1+ mnth\_2+ mnth\_3+ mnth\_4+ mnth\_5+ mnth\_6+ mnth\_7+ mnth\_8+ mnth\_9+ mnth\_10+ mnth\_11+ mnth\_12+ hr\_0+ hr\_1+ hr\_2+ hr\_3+ hr\_4+ hr\_5+ hr\_6+ hr\_7+ hr\_8+ hr\_9+ hr\_10+ hr\_11+ hr\_12+ hr\_13+ hr\_14+ hr\_15+ hr\_16+ hr\_17+ hr\_18+ hr\_19+ hr\_20+ hr\_21+ hr\_22+ hr\_23+ weekday\_6+ weekday\_0+ weekday\_1+ weekday\_2+ weekday\_3+ weekday\_4+ weekday\_5+weathersit\_1+ weathersit\_2+ weathersit\_3+ weathersit\_4, data = train\_data)

**Regression equation:**

-31.149 + 82.242 yr - 5.685 holiday + 12.412 workingday + 120.304 temp + 124.956 atemp - 63.729 hum - 41.947 windspeed - 49.789 season\_1 - 24.384 season\_2 - 13.907 season\_3 - 15.841 mnth\_1 - 11.927 mnth\_2 + 3.175 mnth\_3 + 2.653 mnth\_4 + 15.558 mnth\_5 -4.243 mnth\_6 - 39.440 mnth\_7 - 23.907 mnth\_8 + 1.541 mnth\_9 + 5.445 mnth\_10 - 6.173 mnth\_11 - 26.797 hr\_0 - 40.754 hr\_1 - 47.463 hr\_2 - 56.690 hr\_3 - 59.560 hr\_4 - 45.545 hr\_5 + 3.640 hr\_6 + 114.897 hr\_7 + 238.701 hr\_8 + 115.812 hr\_9 + 66.533 hr\_10 + 87.422 hr\_11 + 120.134 hr\_12 + 117.928 hr\_13 + 103.904 hr\_14 + 111.679 hr\_15 + 162.304 hr\_16 + 293.987 hr\_17 + 268.684 hr\_18 + 174.400 hr\_19 + 104.605 hr\_20 + 64.908 hr\_21 + 34.501 hr\_22 + 10.838 weekday\_6 -10.543 weekday\_1 -7.571 weekday\_2 - 9.381 weekday\_3 - 5.424 weekday\_4 + 57.190 weathersit\_1 + 44.839 weathersit\_2 - 5.597 weathersit\_3

**PART 3:** Implementing the gradient descent algorithm with batch update rule and computing the cost function

**Note:** The code for Cost Function method and Gradient Descent algorithm implementation is attached as part of the R file.

**Initial Parameter values:**

**Betas:** Random float numbers between 0 and 1

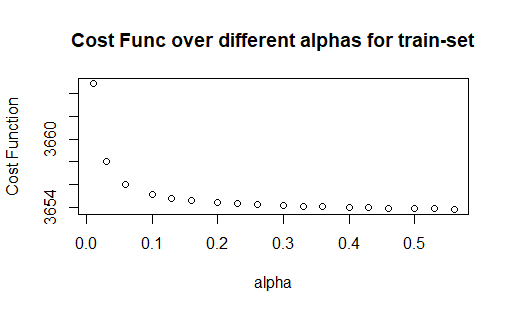
**Learning rate (alpha):** 0.01

**Cost function delta threshold:** 0.1

**EXPERIMENTATION:**

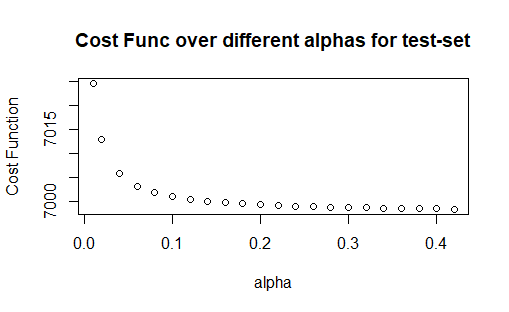
**1. Varying alpha values for training and test data sets.**

Initially the learning rate value is set to **0.01** and been progressively increased till the cost function failed to converge (Cost func. = Infinity value)



For **training data-set**, the cost function reaches its **minimum** value of **3653.848** at **alpha = 0.56** from 3664.866 at alpha = 0.01 and finally reaches Infinity at alpha = 0.6

The **ideal alpha rate** would be around **0.56** as it gives us the minimum cost function value before finally losing to converge.

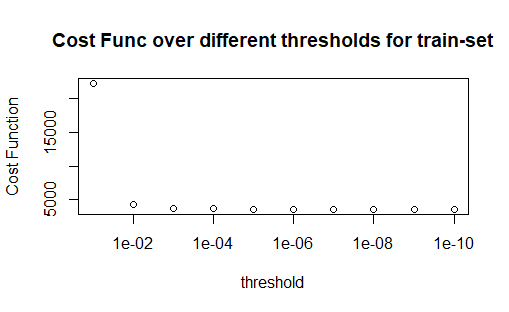


For **testing data-set**, the cost function reaches its **minimum value of 6998.464** at **alpha = 0.42** from 7024.536 at alpha = 0.01 and finally reaches Infinity at alpha = 0.43

The **ideal alpha rate** would be around **0.42** as it gives us the minimum cost function value before finally losing to converge.

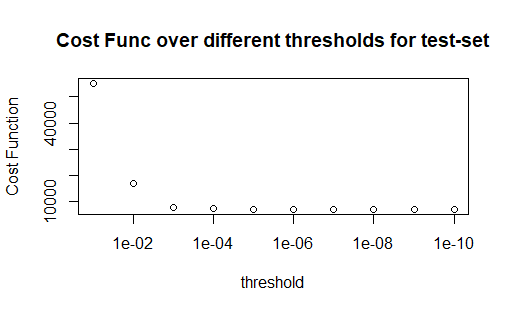
**2. Varying threshold values for training and test data sets.**

Initially the threshold value is set to 0.1 and been progressively increased till the cost function failed to minimize further (reaching its minimum possible value and staying constant with increased iterations)



For **training data-set**, the cost function reaches its **minimum value of 3653.848** at **threshold = 1e-06** from 22155.279 at threshold = 0.1 before settling at a constant value.

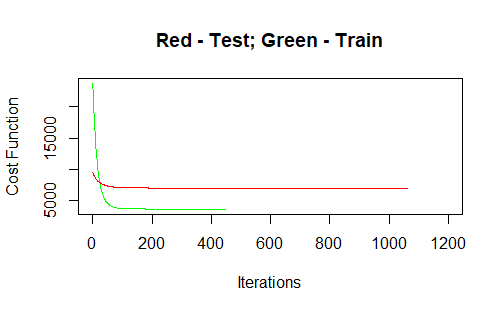
The **ideal threshold rate** would be around **1e-06** (0.000001) for the same reason stated above.



For **testing data-set**, the cost function reaches its **minimum value of 6997.588** at **threshold = 1e-07** from 55080.281 at threshold = 0.1 before settling at a constant value.

The **ideal threshold rate** would be around **1e-07** (0.0000001) for the same reason stated above.

**Plotting train and test errors as a function of number of gradient descent iterations:**



**3.** Three randomly chosen attributes are **season, mnth, weekday.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Cost Function Value** | | | |
| **Train Data set** | | **Test Data set** | |
| **All variables** | **3 random variables** | **All variables** | **3 random variables** |
| **3653.848** | **10824.4814902858** | **6997.588** | **22863.4185023358** |

As we can see from the above results, the cost function value at convergence is much higher for both the train and test data sets of 3-random variable models as opposed to the All-variable model. This proves that **the random model is inefficient**.

**4.** The three cherry-picked features are **hr, temp, weathersit.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Cost Function Value** | | | |
| **Train Data set** | | **Test Data set** | |
| **3 chosen variables** | **3 random variables** | **3 chosen variables** | **3 random variables** |
| **4508.65376457778** | **10824.4814902858** | **7741.47621012051** | **22863.4185023358** |

These chosen features proved to explain/predict the dependent variable **better than those randomly chosen ones**. This is evident from the **lower cost function convergence values** (lowered by about 60%) produced by these features.

|  |  |  |  |
| --- | --- | --- | --- |
| **Cost Function Value** | | | |
| **Train Data set** | | **Test Data set** | |
| **All variables** | **3 chosen variables** | **All variables** | **3 chosen variables** |
| **3653.848** | **4508.65376457778** | **6997.588** | **7741.47621012051** |

**The all-variables model outperformed this cherry-picked feature model by a small margin since it has more explanatory features.**

**Discussion:**

**The final linear regression equation done using gradient descent algorithm with all parameters:**

**26.9047505 + 82.8108614 yr - 6.8786185 holiday + 11.2531240 workingday + 121.8793333 temp + 117.0401528 atemp - 64.4736125 hum - 39.0775359 windspeed - 18.1087756 season\_1 + 6.1117109 season\_2 + 12.3936464 season\_3 + 29.3267964 season\_4 - 11.2621997 mnth\_1 - 6.9598018 mnth\_2 + 9.1091751 mnth\_3 + 9.8417677 mnth\_4 + 23.6662218 mnth\_5 + 6.1676803 mnth\_6 - 25.4455528 mnth\_7 - 10.3406643 mnth\_8 + 13.8663059 mnth\_9 + 14.4715255 mnth\_10 + 2.3975061 mnth\_11 + 7.2564381 mnth\_12 - 104.5872659 hr\_0 - 118.5902422 hr\_1 - 125.2641255 hr\_2 - 134.5056557 hr\_3 - 137.4225100 hr\_4 - 123.4309185 hr\_5 - 74.2737114 hr\_6 + 36.9571346 hr\_7 + 160.7559598 hr\_8 + 37.8427540 hr\_9 - 11.4048074 hr\_10 + 9.5630073 hr\_11 + 42.2871549 hr\_12 + 40.1234013 hr\_13 + 26.1130632 hr\_14 + 33.8974388 hr\_15 + 84.4811297 hr\_16 + 216.1725723 hr\_17 + 190.8165082 hr\_18 + 96.5886076 hr\_19 + 26.8491272 hr\_20 - 12.8289479 hr\_21 - 43.2639760 hr\_22 - 77.7670516 hr\_23 + 17.4327873 weekday\_6 + 6.6560672 weekday\_0 - 2.6880381 weekday\_1 + 0.4270127 weekday\_2 - 1.4426286 weekday\_3 + 2.4746816 weekday\_4 + 7.7476757 weekday\_5 + 34.7919430 weathersit\_1 + 22.6935247 weathersit\_2 - 27.9098806 weathersit\_3 - 0.8527864 weathersit\_4**

**Interpretation:**

1. The intercept value is 26.9047505. This means that **approximately 27 bikes get rented when all the features are not considered** (all feature variables set to zero).
2. Features such as “**workingday**”, “**temp**”, “**atemp**”, “**yr**” shows a **positive impact** on the number of bikes rented (Thanks to their positive coefficient values)
3. Features such as “**holiday**”, “**hum**”, “**windspeed**” shows a **negative impact** on the bike rental count variable due to their negative coefficient values.
4. **Clear, Few clouds and Misty conditions increases** the count value whereas, **light snow or heavy rains go against** the bike rental count.

**Important features:**

Since the requirement is to predict the number of bike rental count given an hour, I think that the variable **“hr” plays a very important role** in predicting the same. Also, the variables “**temp**” and “**weathersit**” contributes more in explaining the dependent variable with **high coefficients**.

**Further steps:**

1. There is a suspect of this model being heteroskedastic (error term might be varying with feature values). We can run a robust standard error linear regression model for better prediction.
2. We can also look for non-linear regression models and see how those fit the data.
3. Finally, it would be better to gather more explanatory variables to better explain the variance in the dependent variable.