## 

ICT707 Big Data Assignment

## Big Data Assignment Marking Criteria

The Big Data Assignment is comprised of two parts:

* The first part is to create the algorithms in the tasks, namely: Decision Tree, Gradient Boosted Tree and Linear regression and then to apply them to the bike sharing dataset provided. Try and produce the output given in the task sections (also given in the **Big-Data Assignment.docx** provided on Blackboard).
* The second part is then use those algorithms created in the first part and apply them to another dataset chosen from Kaggle (other than the bike sharing dataset provided).

# Rubric

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | Datasets | |
| bike sharing [provided] | Student selected dataset [from Kaggle.com] |
| Decision Tree | Decision Tree | | 5 | 5 |
| Decision Tree Categorical features | | 5 | 5 |
| Decision Tree Log | | 5 | 5 |
| Decision Tree Max Bins | | 5 | 5 |
| Decision Tree Max Depth | | 5 | 5 |
| Gradient Boosted Tree | Gradient Boosted Tree | | 5 | 5 |
| Gradient boost tree iterations | | 5 | 5 |
| Gradient boost tree Max Bins | | 5 | 5 |
| Linear regression | Linear regression | | 5 | 5 |
| Linear regression Cross Validation | Intercept | 5 | 5 |
| Iterations | 5 | 5 |
| Step size | 5 | 5 |
| L1 Regularization | 5 | 5 |
| L2 Regularization | 5 | 5 |
| Linear regression Log | | 5 | 5 |
|  |  | | 75 | 75 |
| Total mark | | | | 150 |

# What needs to be submitted for marking:

For the Decision tree section a .py or .ipynb file for each of the following:

* Decision Tree
* Decision Tree Categorical features
* Decision Tree Log
* Decision Tree Max Bins
* Decision Tree Max Depth

For the Gradient boost tree section a .py or .ipynb file for each of the following:

* Gradient boost tree
* Gradient boost tree iterations
* Gradient boost tree Max Bins

For the Linear regression section a .py or .ipynb file for each of the following:

* Linear regression
* Linear regression Cross Validation
* Intercept
* Iterations
* Step size
* L1 Regularization
* L2 Regularization
* Linear regression Log

Each of the files submitted will be tested with the following datasets:

* bike sharing [which is provided on blackboard]
* A dataset of the students choice downloaded from Kaggle.com

[Hint] Write each algorithm so that it can take in a dataset name. For example:

raw\_data = sc.textFile("/home/spark/data/hour.csv")

In this manner both datasets can be run with the same files.

# Assignment

1. Utilising Python 3 Build the following regression models:

* Decision Tree
* Gradient Boosted Tree
* Linear regression

1. Select a dataset (other than the example dataset given in section 3) and apply the Decision Tree and Linear regression models created above. Choose a dataset from Kaggle <https://www.kaggle.com/datasets>
2. Build the following in relation to the gradient boost tree and the dataset choosen in step 2
   1. Gradient boost tree iterations (see **Big-Data Assignment.docx** section 6.1)
   2. Gradient boost tree Max Bins (see **Big-Data Assignment.docx** section 7.2)
3. Build the following in relation to the decision tree and the dataset choosen in step 2
   1. Decision Tree Categorical features
   2. Decision Tree Log (see **Big-Data Assignment.docx**section 5.4)
   3. Decision Tree Max Bins (see **Big-Data Assignment.docx** section 7.2)
   4. Decision Tree Max Depth (see **Big-Data Assignment.docx** section 7.1)
4. Build the following in relation to the linear regression and the dataset choosen in step 2
   1. Linear regression Cross Validation
      1. Intercept (see **Big-Data Assignment.docx** section 6.5)
      2. Iterations (see **Big-Data Assignment.docx** section 6.1)
      3. Step size (see **Big-Data Assignment.docx**section 6.2)
      4. L1 Regularization (see **Big-Data Assignment.docx** section 6.4)
      5. L2 Regularization (see **Big-Data Assignment.docx** section 6.3)
   2. Linear regression Log (see **Big-Data Assignment.docx** section 5.4)
5. Follow the provided example of the Bike sharing data set and the guide lines in the sections that follow this section to develop the requirements given in steps 1,3,4 and 5

## Task 1

Task 1 is comprised of developing:

1. Decision Tree
   1. Decision Tree Categorical features
   2. Decision Tree Log (see **Big-Data Assignment.docx** section 5.4)
   3. Decision Tree Max Bins (see **Big-Data Assignment.docx** section 7.2)
   4. Decision Tree Max Depth (see **Big-Data Assignment.docx** section 7.1)

**The Output for this task and all the sub tasks are based on the the Bike sharing data set as input. Utilise the Bike sharing data set as input to test that the Decision Tree task and sub tasks (i.e.step 1 and 4 from the assignment) are working and producing the correct output before apply to your selected data set.**

### Decision Tree

Output 1:

Feature vector length for categorical features: 57

Feature vector length for numerical features: 4

Total feature vector length: 61

Decision Tree feature vector: [1.0,0.0,1.0,0.0,0.0,6.0,0.0,1.0,0.24,0.2879,0.81,0.0]

Decision Tree feature vector length: 12

Decision Tree predictions: [(16.0, 54.913223140495866), (40.0, 54.913223140495866), (32.0, 53.171052631578945), (13.0, 14.284023668639053), (1.0, 14.284023668639053)]

Decision Tree depth: 5

Decision Tree number of nodes: 63

Decision Tree - Mean Squared Error: 11611.4860

Decision Tree - Mean Absolute Error: 71.1502

Decision Tree - Root Mean Squared Log Error: 0.6251

Output 2:

Decision Tree feature vector: [1.0,0.0,1.0,0.0,0.0,6.0,0.0,1.0,0.24,0.2879,0.81,0.0]

Decision Tree feature vector length: 12

Decision Tree predictions: [(16.0, 54.913223140495866), (40.0, 54.913223140495866), (32.0, 53.171052631578945), (13.0, 14.284023668639053), (1.0, 14.284023668639053)]

Decision Tree depth: 5

Decision Tree number of nodes: 63

Decision Tree - Mean Squared Error: 11611.4860

Decision Tree - Mean Absolute Error: 71.1502

Decision Tree - Root Mean Squared Log Error: 0.6251

### Categorial features

Output:

Mapping of first categorical feature column: {'1': 0, '4': 1, '2': 2, '3': 3}

Categorical feature size mapping {0: 5, 1: 3, 2: 13, 3: 25, 4: 3, 5: 8, 6: 3, 7: 5}

Decision Tree Categorical Features - Mean Squared Error: 7912.5642

Decision Tree Categorical Features - Mean Absolute Error: 59.4409

Decision Tree Categorical Features - Root Mean Squared Log Error: 0.6192

### Decision Tree Log

Output:

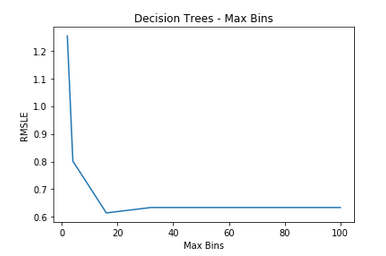
Decision Tree Log - Mean Squared Error: 14781.5760

Decision Tree Log - Mean Absolute Error: 76.4131

Decision Tree Log - Root Mean Squared Log Error: 0.6406

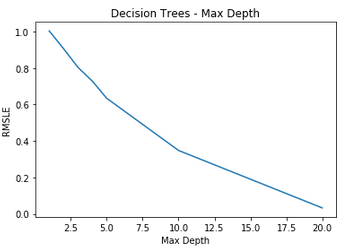
### Decision Tree Max Bins

Output:



### Decision Tree Max Depth

Output:



## Task 2

Task 2 is compromised of developing:

1. Gradient boost tree
   1. Gradient boost tree iterations (see **Big-Data Assignment.docx** section 6.1)
   2. Gradient boost tree Max Bins (see **Big-Data Assignment.docx**section 7.2)
   3. Gradient boost tree Max Depth (see **Big-Data Assignment.docx** section 7.1)

### Gradient Boosted Tree

Output:

GradientBoosted Trees predictions: [(16.0, 103.33972087713495), (40.0, 103.33972087713495), (32.0, 103.33972087713495), (13.0, 103.33972087713495), (1.0, 103.33972087713495)]

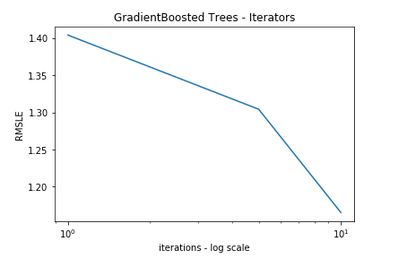
Gradient Boosted Trees - Mean Squared Error = 325939579.98366314

Gradient Boosted Trees - Mean Absolute Error = 1845603.969

Gradient Boosted Trees - Mean Root Mean Squared Log Error = 32155.5757154

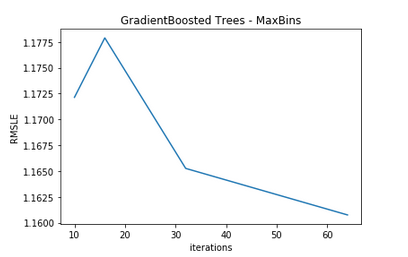
### Gradient boost tree iterations

Output:



### Gradient boost tree Max Bins

Output:



## Task 3

Task 3 is compromised of developing:

1. Linear regression model
   1. Linear regression Cross Validation
      1. Intercept (see **Big-Data Assignment.docx** section 6.5)
      2. Iterations (see **Big-Data Assignment.docx** section 6.1)
      3. Step size (see **Big-Data Assignment.docx** section 6.2)
      4. L1 Regularization (see **Big-Data Assignment.docx** section 6.4)
      5. L2 Regularization (see **Big-Data Assignment.docx** section 6.3)
   2. Linear regression Log (see **Big-Data Assignment.docx** section 5.4)

### Linear regression model

Output:

Mapping of first categorical feature column: {'1': 0, '4': 1, '2': 2, '3': 3}

Feature vector length for categorical features: 57

Feature vector length for numerical features: 4

Total feature vector length: 61

Linear Model feature vector:

[1.0,0.0,0.0,0.0,1.0,0.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,0.0,0.24,0.2879,0.81,0.0]

Linear Model feature vector length: 61

Gradient Boosted Trees - Mean Root Mean Squared Log Error = 32155.5757154

Output 2:

Linear Model predictions: [(16.0, 53.183375554478182), (40.0, 52.572149013454187), (32.0, 52.517786871472346), (13.0, 52.312352839640027), (1.0, 52.285323002218234)]

Linear Regression - Mean Squared Error: 46565.6666

Linear Regression - Mean Absolute Error: 148.3472

Linear Regression - Root Mean Squared Log Error: 1.4284

### Linear regression Cross Validation

Output:

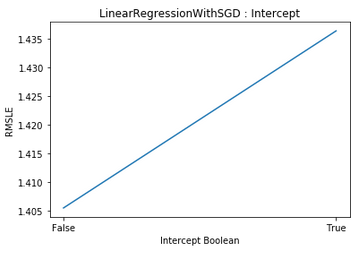
Training data size: 13869

Test data size: 3510

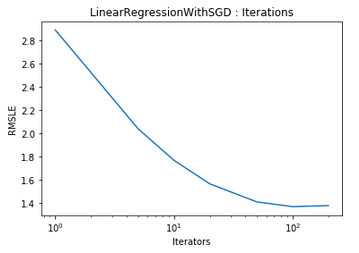
Total data size: 17379

Train + Test size : 17379

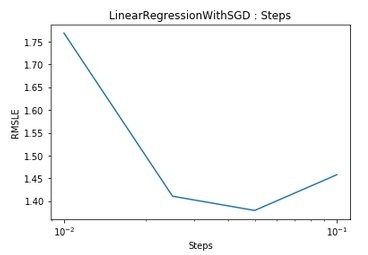
#### Intercept

Output:

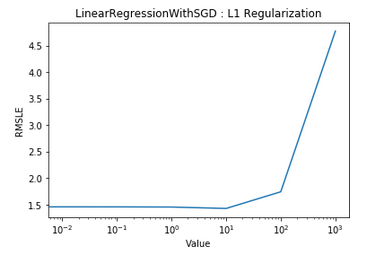
#### Iterations

Output:

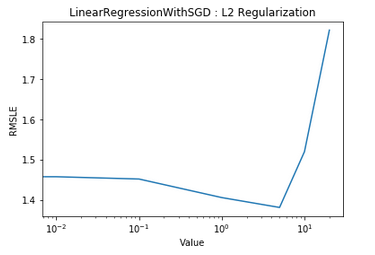
#### Step size

Output:

#### L1 Regularization

Output:

#### L2 Regularization

Output:

### Linear regression Log

Output:

Linear Regression Log - Mean Squared Error: 50685.5559

Linear Regression Log - Mean Absolute Error: 155.2955

Linear Regression Log - Root Mean Squared Log Error: 1.5411