**A ToolBox for Linear Model in Julia Tensorflow (LMTF)**

**Instructions**

* Set the current working directory as : cd(raw"/home/alok/Desktop/Blackcoffer/Julia+Tensor Flow/Data")
* Save LMTF.jl in same working directory.
* Import it using : include(“LMTF.jl”)
* Import the train data in Train as : Train = CSV.read("trainSet.csv");
* Create a dataframe X\_train of independent variable indicating categorical variable as categorical as : X\_train = DataFrame(YEARMADERANGE = (Train[:YEARMADERANGE]), WALLTYPE = categorical(Train[:WALLTYPE]),................., WSF = (Train[:WSF]) );
* Define the dependent variable in Y\_train as : Y\_train = Train[:, filter(x -> x == :KWH, names(Train))]; ##Here KWH is dependent variable
* Import the test data in Test to test the model as : Test = CSV.read("testSet.csv");
* Define X\_test and Y\_test for test set too.

X\_test = Test[:, filter(x -> x != :KWH, names(Test))];

Y\_test = Test[:, filter(x -> x == :KWH, names(Test))];

* Set the number of significant variable you want in the model as varcount : varcount = 18. As default the model will show all the variables if no value is passed.
* Set the value of alpha for gradient descent optimizer as : alpha = 0.0000000001

Default value of alpha is 0.0000000001.

* Call the function LMTF as

LMTF (X\_train, Y\_train, X\_test, Y\_test);

LMTF (X\_train, Y\_train, X\_test, Y\_test, varcount=18);

LMTF (X\_train, Y\_train, X\_test, Y\_test, varcount=18, alpha=0.00001);

as required for the dataset

* Weights, Bais, t\_stat and sum of square error on test set will be displayed after the iteration is complete.
* To see weights, bais, t\_stat, loss individually you can use curr\_W, curr\_b, t\_stat, curr\_loss after the process is complete.
* Wait for the output file with predicted result which will be saved as Resultv4.csv in the current directory folder.

\*\*\*\*\* open “LMTF\_use\_example.jl” file and see the complete code for “trainSet.csv” file