```
library(mxnet)
library(Metrics)

start.time <- Sys.time()
hr_data <-  read.csv("enbcsv.csv")
trainfile <- hr_data[1:670,1:10]
write.csv(trainfile,"enbtrain2.csv")
testfile <- hr_data[671:768,1:10]
write.csv(testfile,"enbtest2.csv")

train <- data.matrix(trainfile)
test = data.matrix(testfile)

train.x <- train[,-9]
train.y <- train[,9]
test.x <- test[,-9]
test.y <- test[,9]
train.x1 <- train[,-10]
train.y1 <- train[,10]
test.x1 <- test[,-10]
test.y1 <- test[,10]

data <- mx.symbol.Variable("data")
fc1 <- mx.symbol.FullyConnected(data, num_hidden=1)

# Use linear regression for the output layer
lro <- mx.symbol.LinearRegressionOutput(fc1)
mx.set.seed(0)
model <- mx.model.FeedForward.create(lro, X=train.x, y=train.y,
                                     ctx=mx.cpu(), num.round=50, array.batch.size=20,
                                     learning.rate=2e-6, momentum=0.9, eval.metric=mx.metric.rmse)
```

```
model1 <- mx.model.FeedForward.create(lro, X=train.x1, y=train.y1,
                                      ctx=mx.cpu(), num.round=50, array.batch.size=20,
                                      learning.rate=2e-6, momentum=0.9, eval.metric=mx.metric.rmse)

preds = predict(model, test.x)
preds

preds1 = predict(model1, test.x1)
preds1

cbindin <- cbind(
  preds,
  preds1)
cbindin


## Auto detect layout of input matrix, use rowmajor..
c1 <- cbind(
  mse<-mean(test.y-preds)^2,

  mse1<-mean(test.y1-preds1)^2)
c1
c2 <- cbind(
  rmse<-sqrt(mse),
  rmse1<-sqrt(mse1))
c2

tss <- sum((test.y - mean(test.y)) ^ 2)  ## total sum of squares

regss <- sum((test.y - preds) ^ 2) ## regression sum of squares
```

```
## Auto detect layout of input matrix, use rowmajor..


tss1 <- sum((test.y1 - mean(test.y1)) ^ 2)  ## total sum of squares


regss1 <- sum((test.y1 - preds1) ^ 2) ## regression sum of squares
c3 <- cbind(
  1-regss / tss,
  1-regss1 / tss1)
c3
rbind(c1,c2,c3)
end.time <- Sys.time()
time.taken <- end.time - start.time
time.taken
```