CS539: Machine Learning

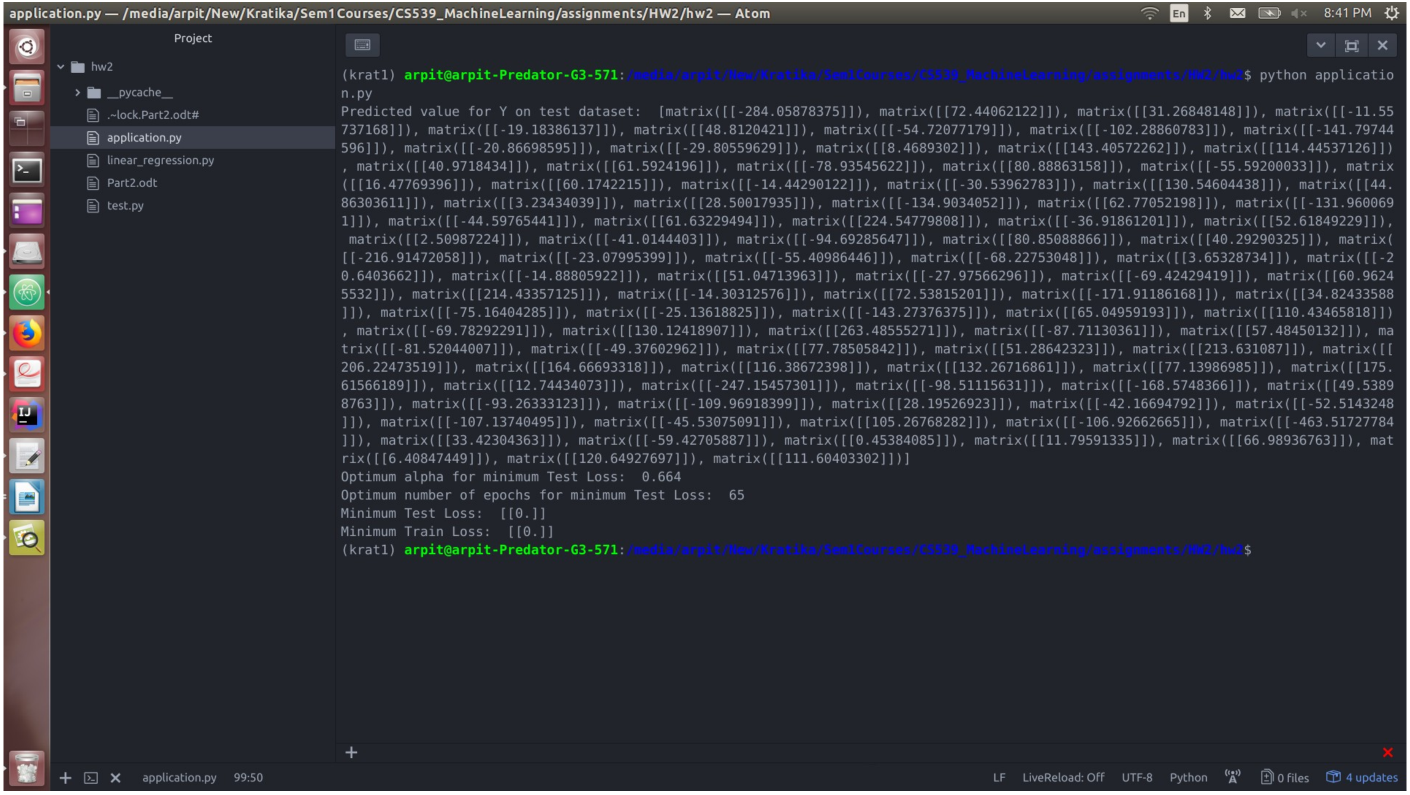
HW2 – Linear Regression

Submitted by: Kratika Agrawal

Part 2: Make predictions by using your implementation

- Code is available in application.py file.
- To calculate optimum value for alpha, so that test losses are minimum, I've computed in function **get_alpha(Xtest,Ytest,Xtrain,Ytrain)**, keeping number of epochs constant at 500.
  Here, **Optimum_alpha = 0.664** => Minimum_TestLoss = 0. (less than 0.001)
- To calculate optimum value for number of epochs. So that test losses are minimum, I've computed in function **get_epochs(Xtest,Ytest,Xtrain,Ytrain,alpha)** passing the optimum value of alpha (0.664) calculated through get_alpha function.
  Here, **Optimum_epochs = 65** => Minimum_TestLoss = 0. (less that 0.001), Minimum_TrainLoss = 0. (less than 0.001)

- Predicted value of Y (Yhat) for test samples:



```
(krat1) arpit@arpit-Predator-G3-571:/media/arpit/New/Kratika/Sem1Courses/CS539_MachineLearning/assignments/HW2/hw2$ python applicatio
n.py
Predicted value for Y on test dataset:  [matrix([[-284.05878375]]), matrix([[72.44062122]]), matrix([[31.26848148]]), matrix([[-11.55
737168]]), matrix([[-19.18386137]]), matrix([[48.8120421]]), matrix([[-54.72077179]]), matrix([[-102.28860783]]), matrix([[-141.79744
596]]), matrix([[-20.86698595]]), matrix([[-29.80559629]]), matrix([[8.4689302]]), matrix([[143.40572262]]), matrix([[114.44537126]])
, matrix([[40.9718434]]), matrix([[61.5924196]]), matrix([[-78.93545622]]), matrix([[80.88863158]]), matrix([[-55.59200033]]), matrix
([[16.47769396]]), matrix([[60.1742215]]), matrix([[-14.44290122]]), matrix([[-30.53962783]]), matrix([[130.54604438]]), matrix([[44.
86303611]]), matrix([[3.23434039]]), matrix([[28.50017935]]), matrix([[-134.9034052]]), matrix([[62.77052198]]), matrix([[-131.960069
1]]), matrix([[-44.59765441]]), matrix([[61.63229494]]), matrix([[224.54779808]]), matrix([[-36.91861201]]), matrix([[52.61849229]]),
 matrix([[2.50987224]]), matrix([[-41.0144403]]), matrix([[-94.69285647]]), matrix([[80.85088866]]), matrix([[40.29290325]]), matrix(
[[-216.91472058]]), matrix([[-23.07995399]]), matrix([[-55.40986446]]), matrix([[-68.22753048]]), matrix([[3.65328734]]), matrix([[-2
0.6403662]]), matrix([[-14.88805922]]), matrix([[51.04713963]]), matrix([[-27.97566296]]), matrix([[-69.42429419]]), matrix([[60.9624
5532]]), matrix([[214.43357125]]), matrix([[-14.30312576]]), matrix([[72.53815201]]), matrix([[-171.91186168]]), matrix([[34.82433588
]]), matrix([[-75.16404285]]), matrix([[-25.13618825]]), matrix([[-143.27376375]]), matrix([[65.04959193]]), matrix([[110.43465818]])
, matrix([[-69.78292291]]), matrix([[130.12418907]]), matrix([[263.48555271]]), matrix([[-87.71130361]]), matrix([[57.48450132]]), ma
trix([[-81.52044007]]), matrix([[-49.37602962]]), matrix([[77.78505842]]), matrix([[51.28642323]]), matrix([[213.631087]]), matrix([[
206.22473519]]), matrix([[164.66693318]]), matrix([[116.38672398]]), matrix([[132.26716861]]), matrix([[77.13986985]]), matrix([[175.
61566189]]), matrix([[12.74434073]]), matrix([[-247.15457301]]), matrix([[-98.51115631]]), matrix([[-168.5748366]]), matrix([[49.5389
8763]]), matrix([[-93.26333123]]), matrix([[-109.96918399]]), matrix([[28.19526923]]), matrix([[-42.16694792]]), matrix([[-52.5143248
]]), matrix([[-107.13740495]]), matrix([[-45.53075091]]), matrix([[105.26768282]]), matrix([[-106.92662665]]), matrix([[-463.51727784
]]), matrix([[33.42304363]]), matrix([[-59.42705887]]), matrix([[0.45384085]]), matrix([[11.79591335]]), matrix([[66.98936763]]), mat
rix([[6.40847449]]), matrix([[120.64927697]]), matrix([[111.60403302]])]
Optimum alpha for minimum Test Loss:  0.664
Optimum number of epochs for minimum Test Loss:  65
Minimum Test Loss:  [[0.]]
Minimum Train Loss:  [[0.]]
(krat1) arpit@arpit-Predator-G3-571:/media/arpit/New/Kratika/Sem1Courses/CS539_MachineLearning/assignments/HW2/hw2$
```

- The relation between alpha and number of epochs is computed in function **get_alpha_epochs_relation(Xtrain,Ytrain,Xtest,Ytest)**. Here, alpha is iterated from 1 to 0.1 and number of epochs are iterated from 10 to 100. Training loss and test losses are calculated for each combination.
  The CSV file that depicts the relation Part2.csv is attached alongwith (as generated by get_alpha_epochs_relation function).
  For this scenario, it signifies that given a value for alpha, test loss decreases with increased number of epochs. And given a value of epoch, test loss increases on decreasing value of alpha.