**Section A.**

**a.**

**b.** No it does not imply that. To Illustrate this let’s take the following example –

X: Outcome of rolling dice 1

Y: Outcome of rolling dice 2

Z: Sum of X and Y

X and Z are correlated and so are Y and Z however X and Y have no correlation!

In general case some more examples can be –

X: Random Variable 1

Y: Random Variable 2

Z: X + Y or max (X,Y) or min(X,Y) or X-Y etc.

**c.**

**d.**

**Section B.**

a.

For all training runs model was set to run for 1000 epochs however early stopping with patience 5 and min\_delta 1e-5 over val\_loss was added to prevent overfitting.

Data was scaled using a custom implementation of Min-Max Scaler

Average Validation RMSE is used to highlight best value of K in each of the following tables:

**Learning rate = 0.0001**

|  |  |  |  |
| --- | --- | --- | --- |
| K (Number of Folds) | Average Train RMSE | Average Validation RMSE | Average Test RMSE |
| 2 | 36.46706212406853 | 36.46445738264513 | 39.122390473948286 |
| 3 | 36.45438303485597 | 36.41266003031339 | 36.49337892890277 |
| 4 | 36.467878026029965 | 36.433046302719426 | 34.2820847308869 |
| 5 | 36.474862060043606 | 36.45290192972408 | 37.22749170997242 |

**Learning rate = 0.001**

|  |  |  |  |
| --- | --- | --- | --- |
| K (Number of Folds) | Average Train RMSE | Average Validation RMSE | Average Test RMSE |
| 2 | 19.11188832202627 | 19.122484362752534 | 20.075953310149785 |
| 3 | 19.126110509331067 | 19.107525621824482 | 18.451620517778178 |
| 4 | 19.07590896900205 | 18.981380132750047 | 19.085228642628444 |
| 5 | 19.131680374291264 | 19.093621308687535 | 19.939762652723523 |

**Learning rate = 0.01**

|  |  |  |  |
| --- | --- | --- | --- |
| K (Number of Folds) | Average Train RMSE | Average Validation RMSE | Average Test RMSE |
| 2 | 13.572939263660144 | 13.604130977214616 | 13.01017232733161 |
| 3 | 13.628025046568638 | 13.631580583103158 | 14.277562510913363 |
| 4 | 13.658533896207867 | 13.629660412770006 | 13.610418519623522 |
| 5 | 13.60702860156988 | 13.51456430851855 | 11.731268253457348 |

**Learning rate = 0.1**

|  |  |  |  |
| --- | --- | --- | --- |
| K (Number of Folds) | Average Train RMSE | Average Validation RMSE | Average Test RMSE |
| 2 | 13.530204029103361 | 13.539130243278821 | 12.355699492362923 |
| 3 | 13.561122134843764 | 13.523076542062753 | 12.325650131596966 |
| 4 | 13.55269947495478 | 13.476467611728992 | 12.795220693270343 |
| 5 | 13.632592986696059 | 13.511909563232644 | 13.341835153099453 |

**Learning rate = 1**

|  |  |  |  |
| --- | --- | --- | --- |
| K (Number of Folds) | Average Train RMSE | Average Validation RMSE | Average Test RMSE |
| 2 | 13.51630833733965 | 13.539200984811364 | 11.412294621856624 |
| 3 | 13.603075599825031 | 13.586079470754589 | 18.477494482898624 |
| 4 | 13.595355003631111 | 13.591208401176965 | 9.781779460069849 |
| 5 | 13.56507579933685 | 13.528138836680807 | 13.381557330166842 |

**Learning rate = 10 (Meaningless Results as Learning Rate too High and not possible to make any meaningful Conclusions as Model Parameters have Exploded)**

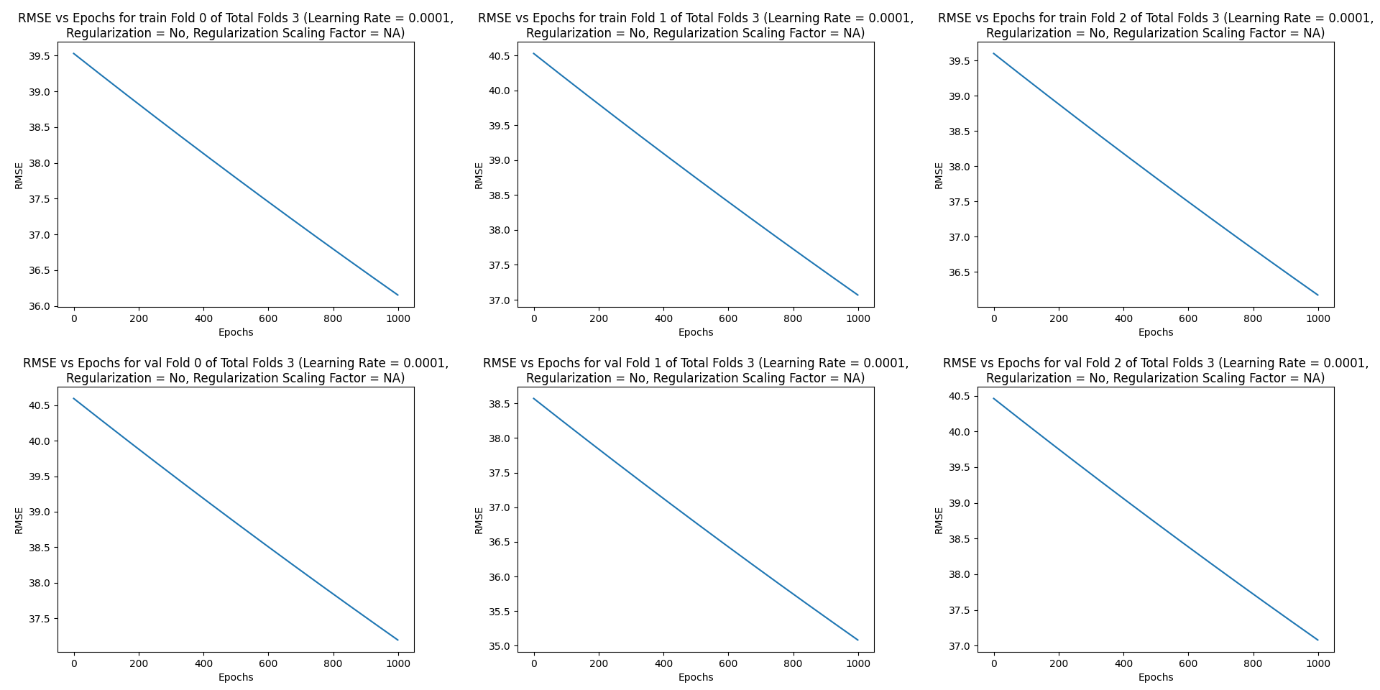
|  |  |  |  |
| --- | --- | --- | --- |
| K (Number of Folds) | Average Train RMSE | Average Validation RMSE | Average Test RMSE |
| 2 | 2602338.0892727086 | 24183651.204742163 | 24208986.611842863 |
| 3 | 2596814.795496345 | 24132817.75013713 | 24081920.2314418 |
| 4 | 2599008.734584617 | 24152633.661489174 | 24096268.700142633 |
| 5 | 2598318.8861544137 | 24144558.378245533 | 24207654.108725123 |

Initial experimentation also included smaller learning rates however it was noticed that the model doesn’t learn anything and hence they’ve been excluded from subsequent analysis.

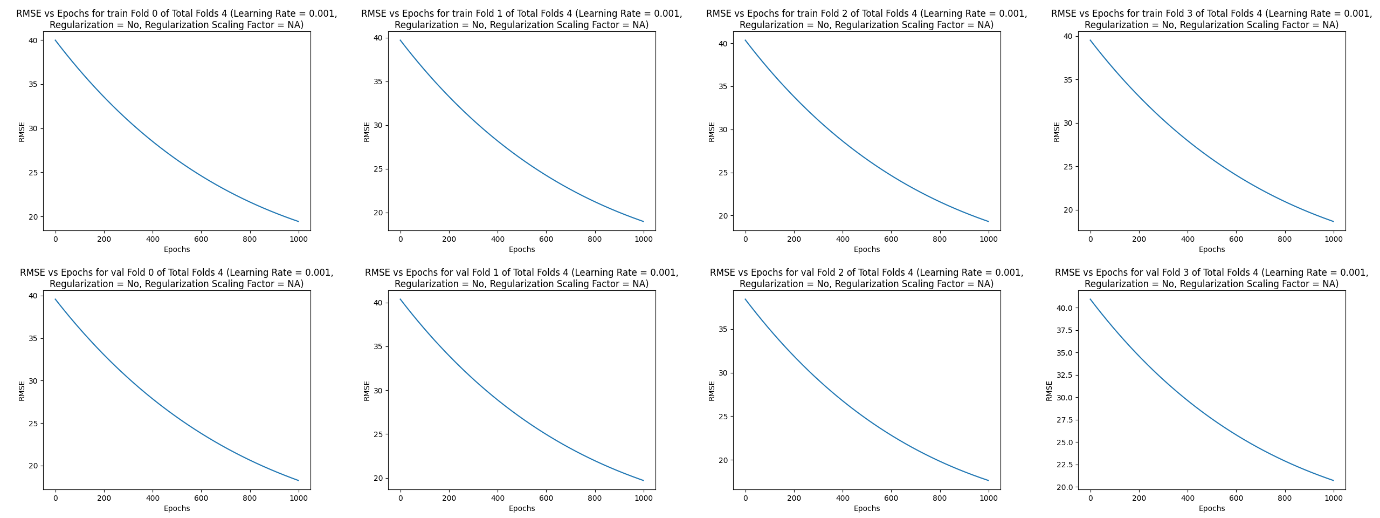
Overall, K=4 seems like the best choice!

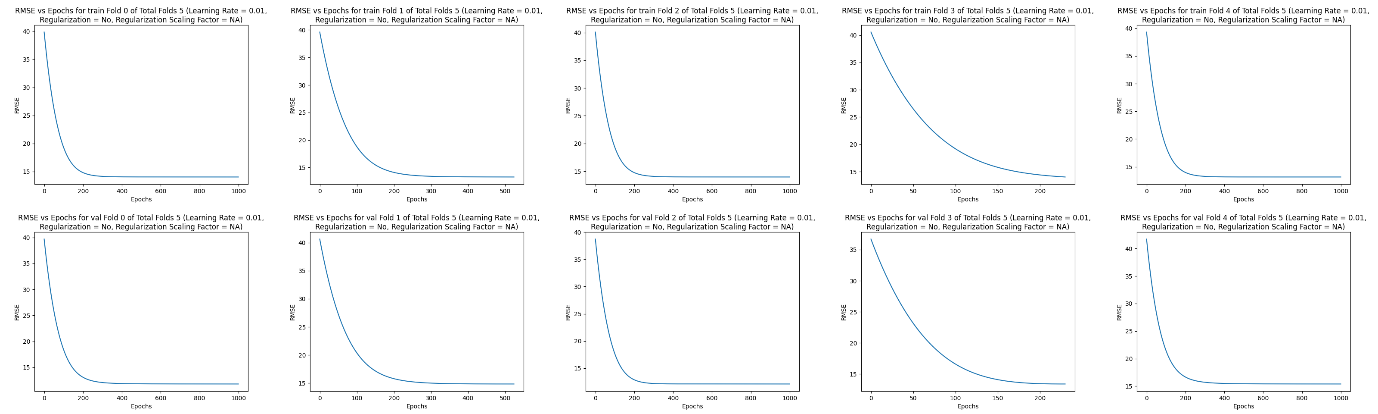
Some plots for best K for each Learning Rate:

Learning Rate = 0.0001 (3-Fold)

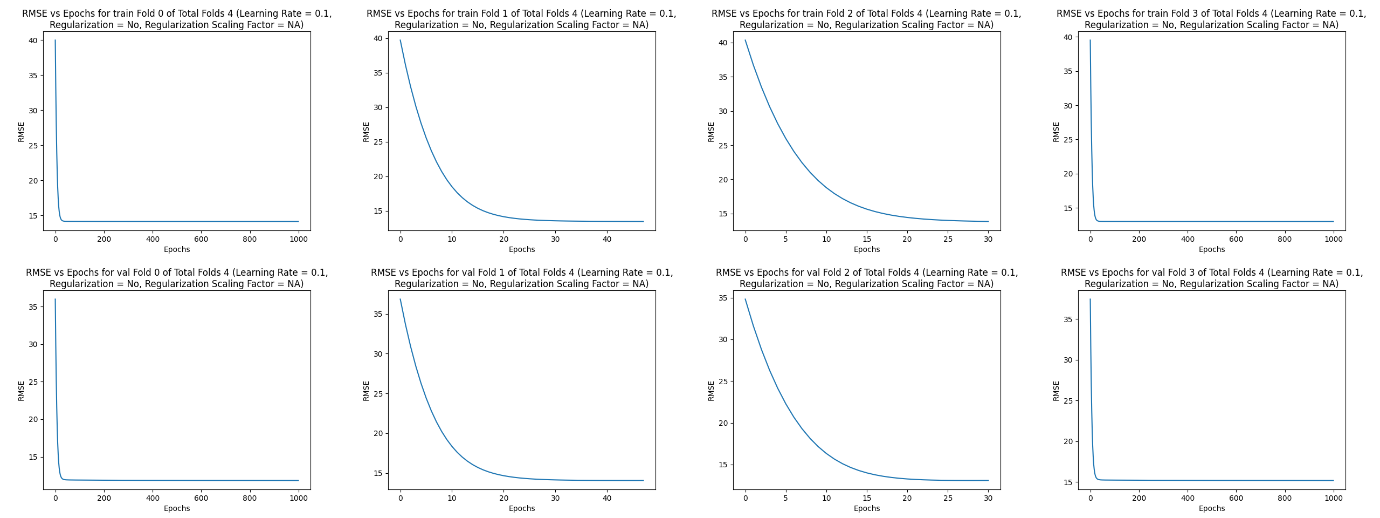


Learning Rate = 0.001 (4-Fold)

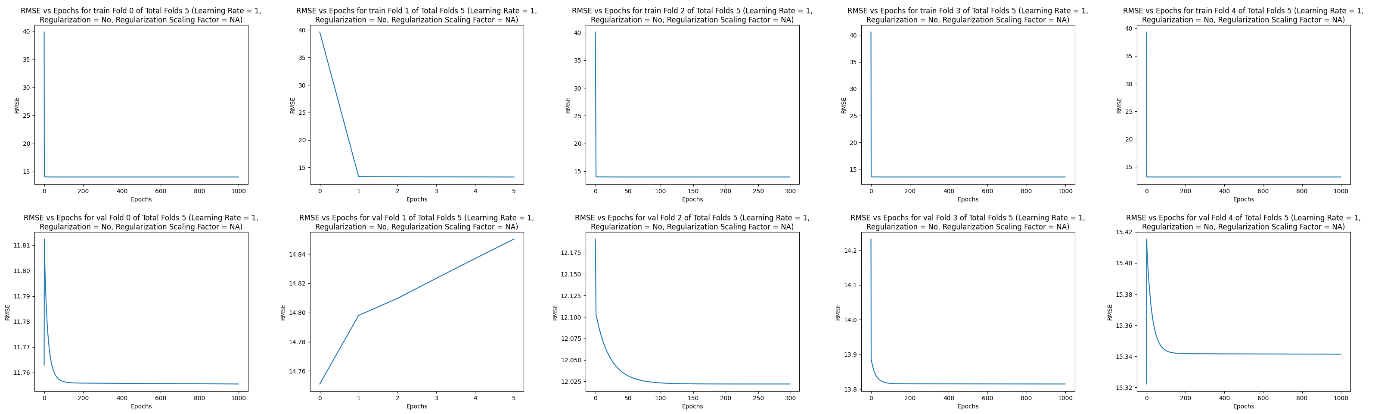


Learning Rate = 0.01 (5-Fold) 

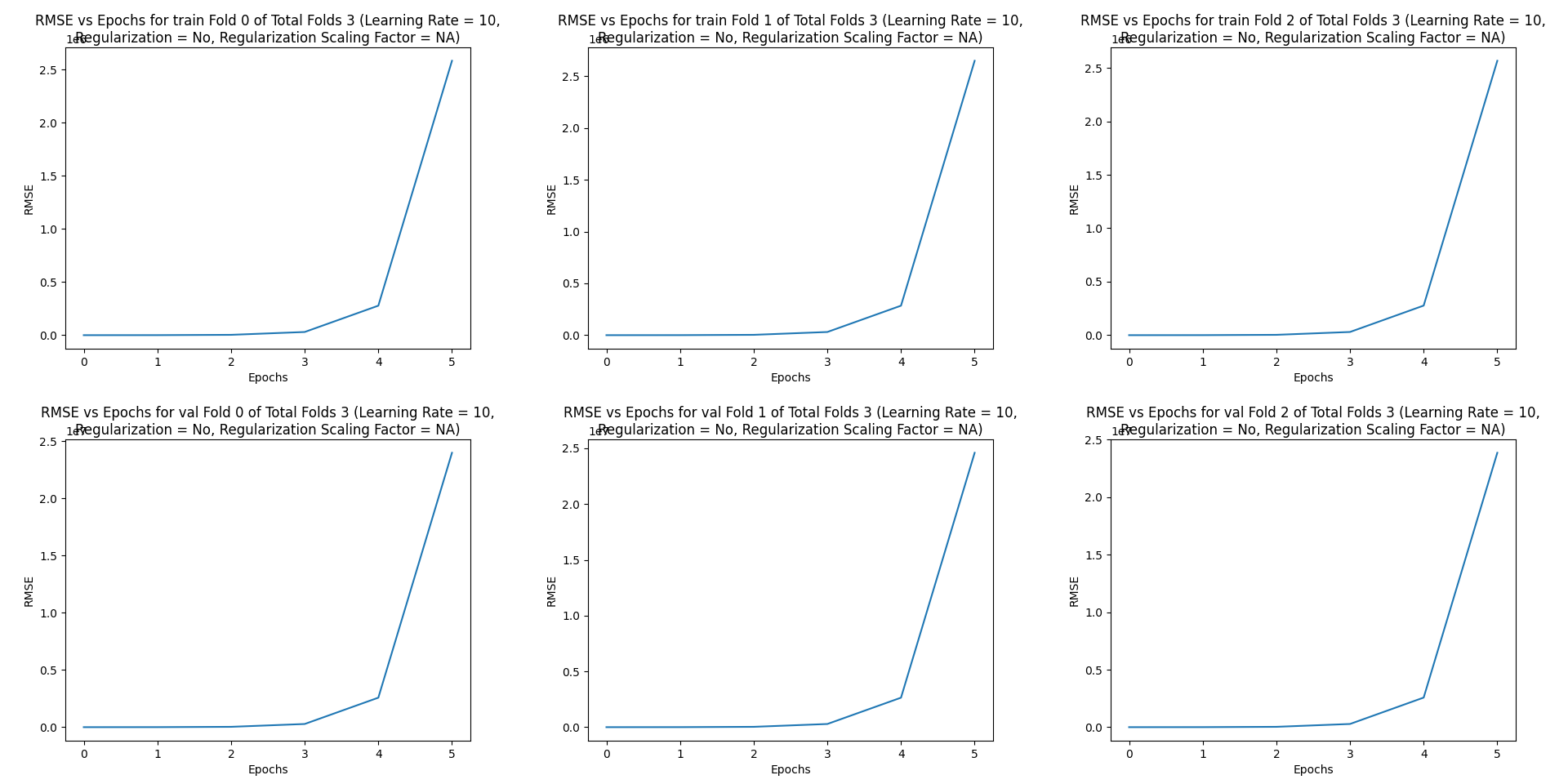
Learning Rate = 0.1 (4-Fold)



Learning Rate = 1 (5-Fold)



Learning Rate = 10 (3-Fold)



b.

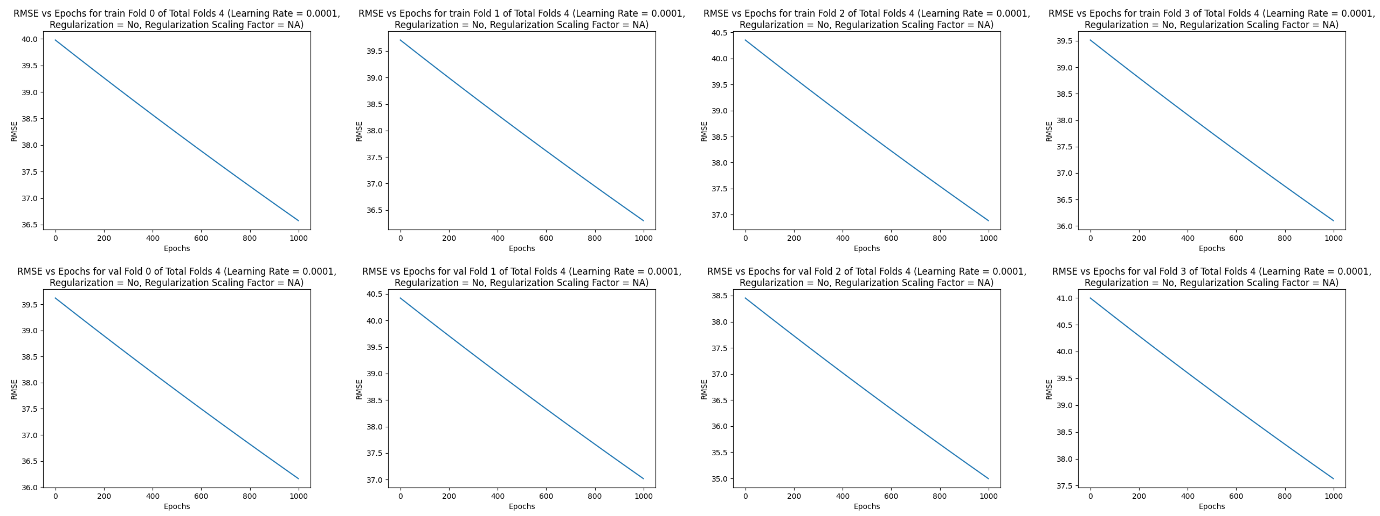
Optimal Number of Folds = 4

**Learning rate = 0.0001:**

Average Train RMSE: 36.46604095920373,

Average Validation RMSE: 36.45175977730566,

Average Testing RMSE: 39.12242116683955

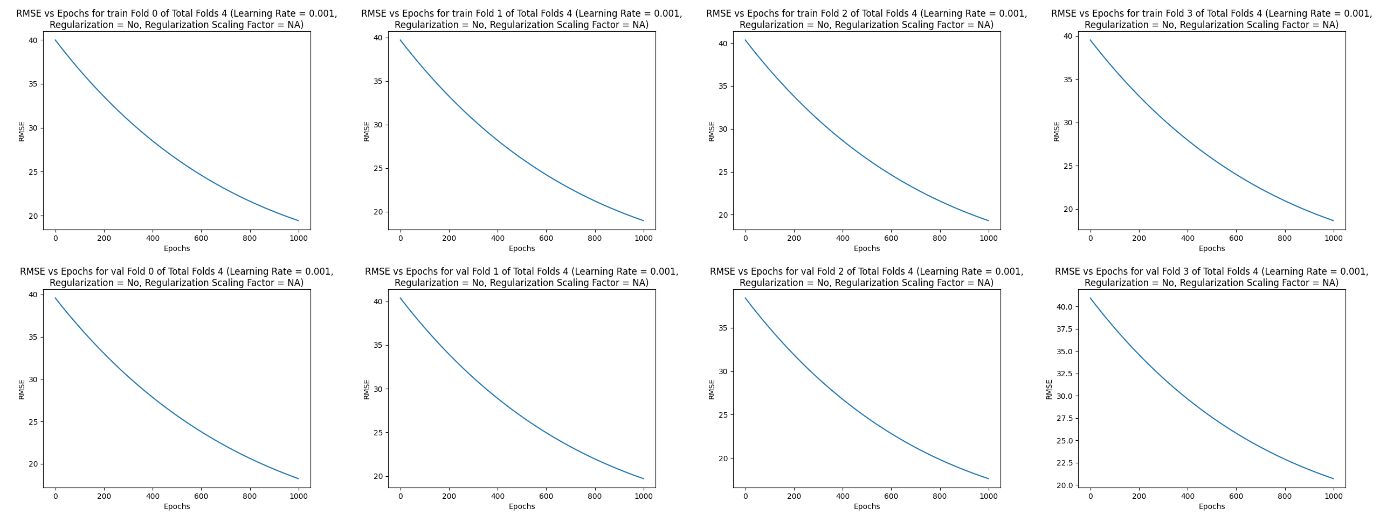


**Learning Rate = 0.001**

Average Train RMSE: **19.10178555917517**,

Average Validation RMSE: **19.075166202706967,**

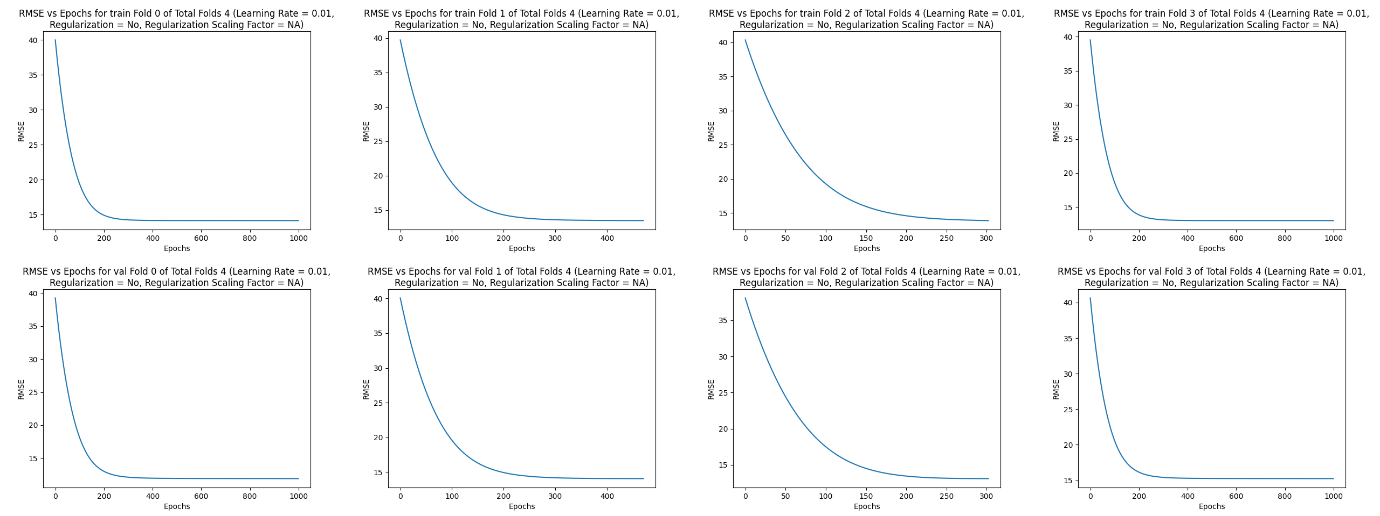
Average Testing RMSE: **21.668222006357805**



**Learning Rate = 0.01**

Average Train RMSE: 13.62103243713321,

Average Validation RMSE: 13.573719074294818**,**

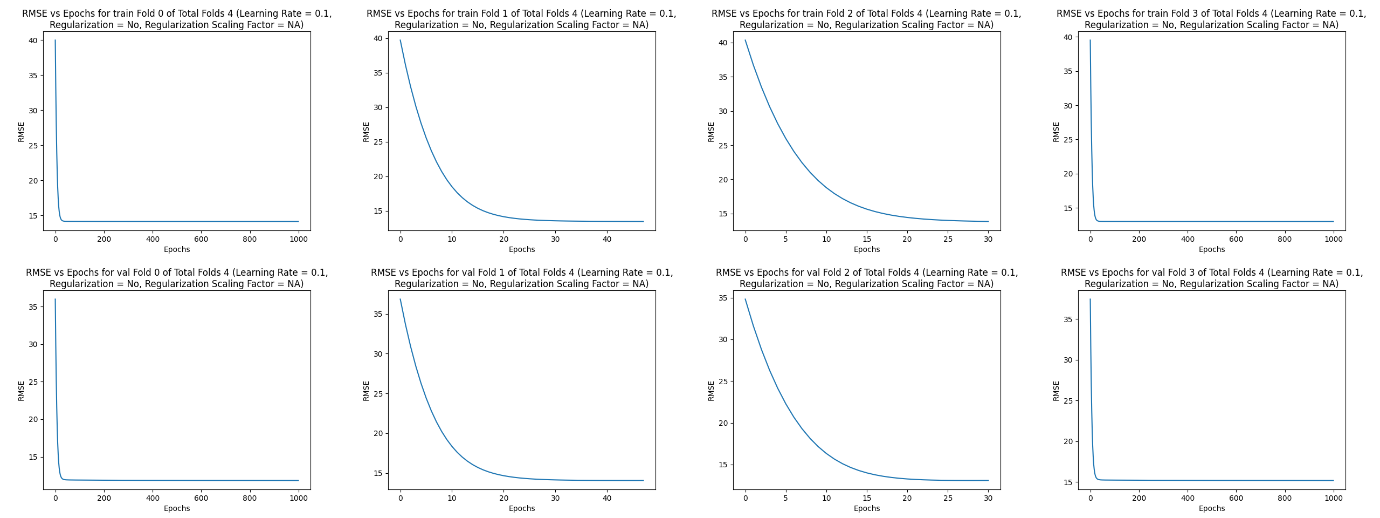
Average Testing RMSE: 15.36892042029418

**Learning Rate = 0.1**

Average Train RMSE: 13.610605543982654,

Average Validation RMSE: 13.55093176612437**,**

Average Testing RMSE: 15.320848075394563

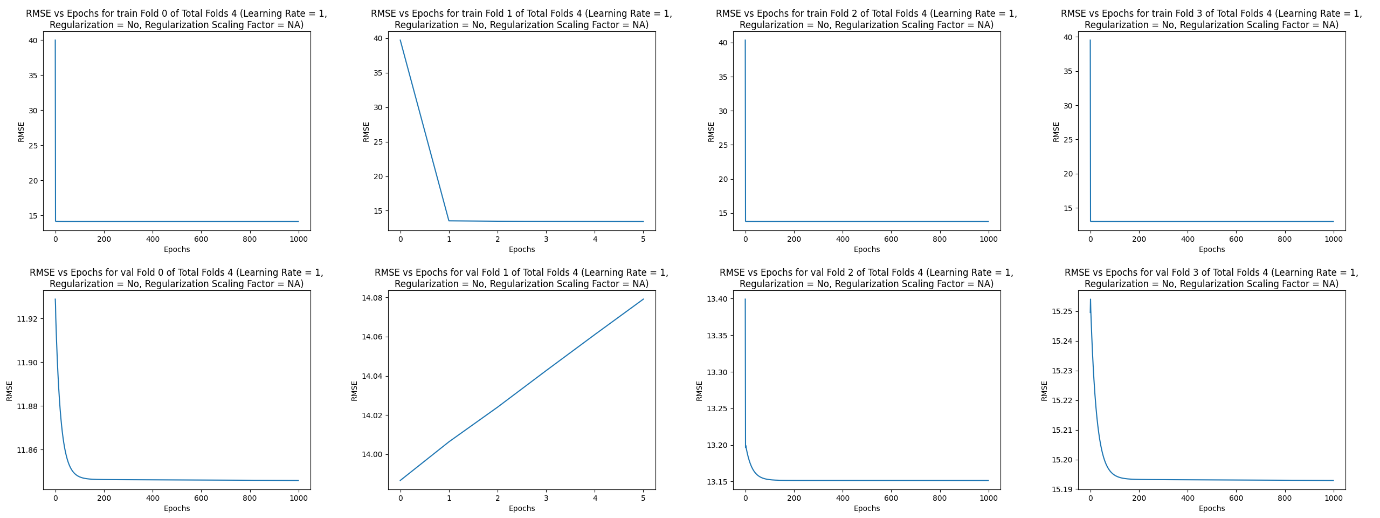


**Learning Rate = 1**

Average Train RMSE: 13.589466507022072,

Average Validation RMSE: 13.55093176612437**,**

Average Testing RMSE: 15.320848075394563



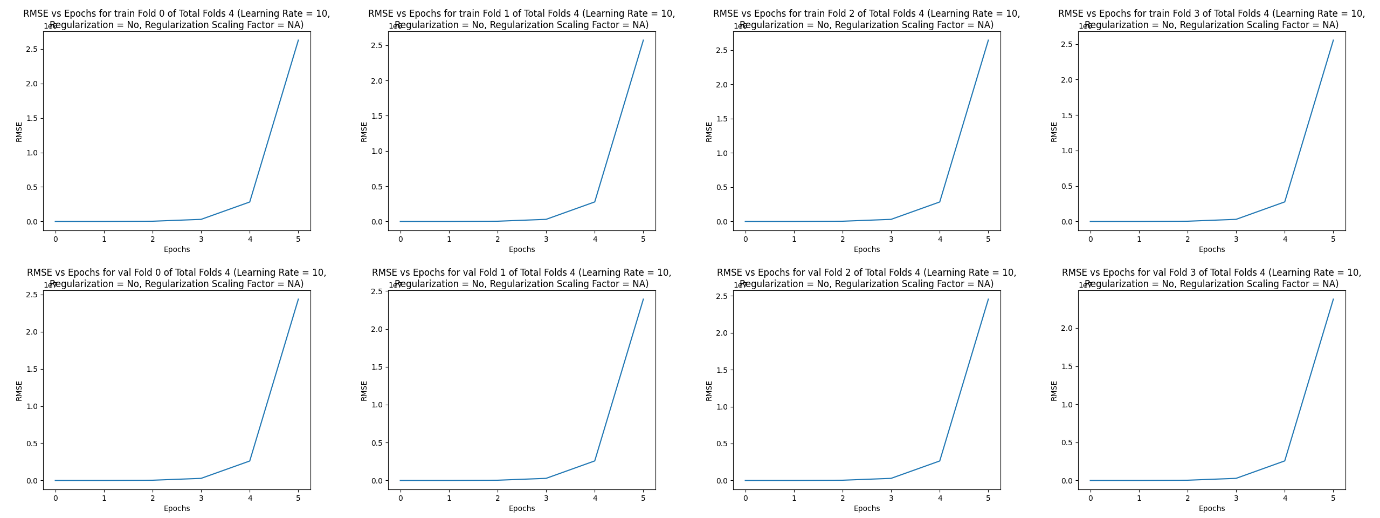
Observation: It seems Fold 1 is an unlucky split for learning rate 1 and hence model performs worse on it as it combines with the fact that we have an unusually high learning rate

**Learning Rate = 10 (The Results Show No Convergence Due to High L.R.)**

Average Train RMSE: 2599704.437786403,

Average Validation RMSE: 24157503.362629715,

Average Testing RMSE: 24363464.19916731



From the plots above a learning rate of 0.1 and 0.01 both seem like a good choice. With learning rate 0.1 we can see in some plots convergence is quicker however that also makes me concerned that this may induce a higher chance of overshooting the local minima. Hence for all subsequent studies I’ve used a learning rate of 0.01

c.

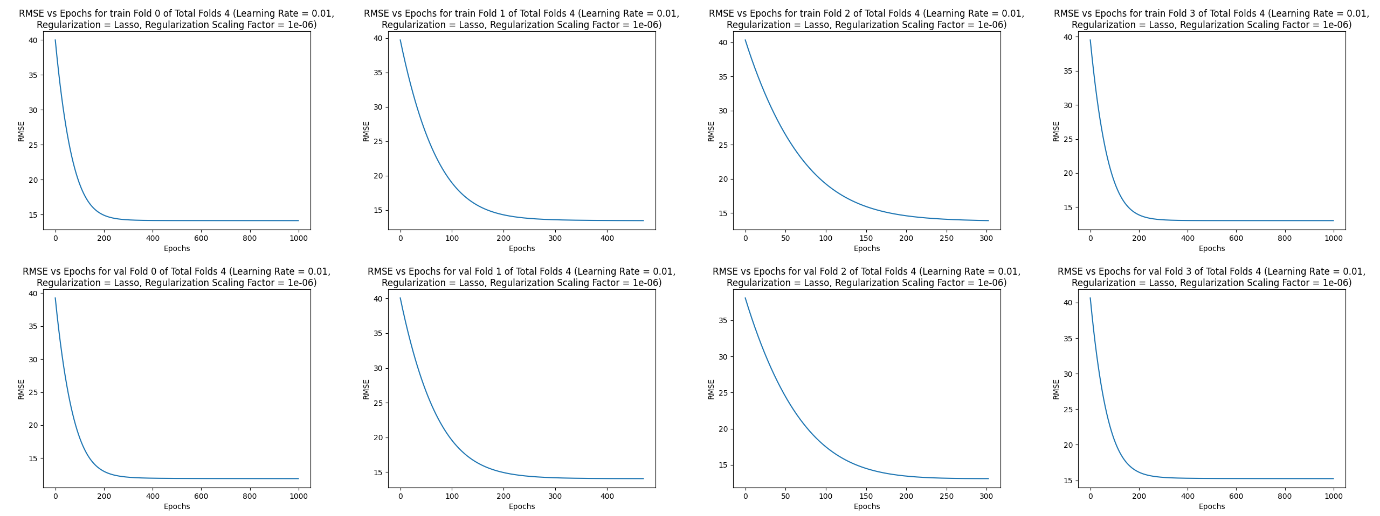
**Lasso Regression**

Regularization Scaling Factor: 1e-6

Average Train RMSE: 13.621032437103091,

Average Validation RMSE: 13.573719074220254,

Average Testing RMSE: 15.368920420477126

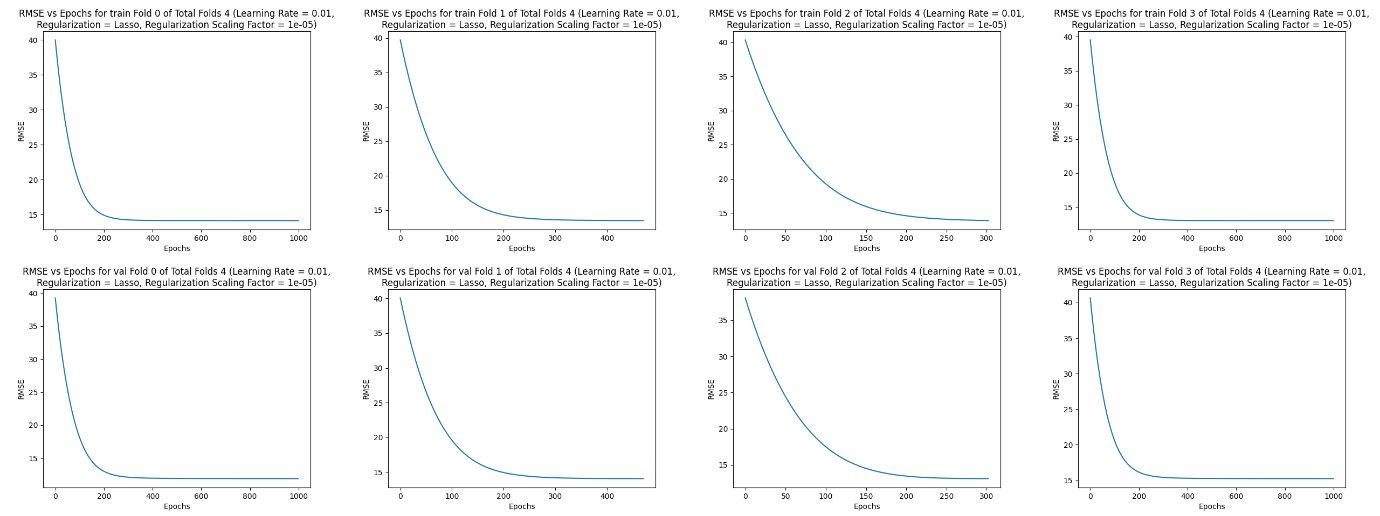


Regularization Scaling Factor: 1e-5

Average Train RMSE: 13.621032436832028,

Average Validation RMSE: 13.573719073549158,

Average Testing RMSE: 15.368920422123637

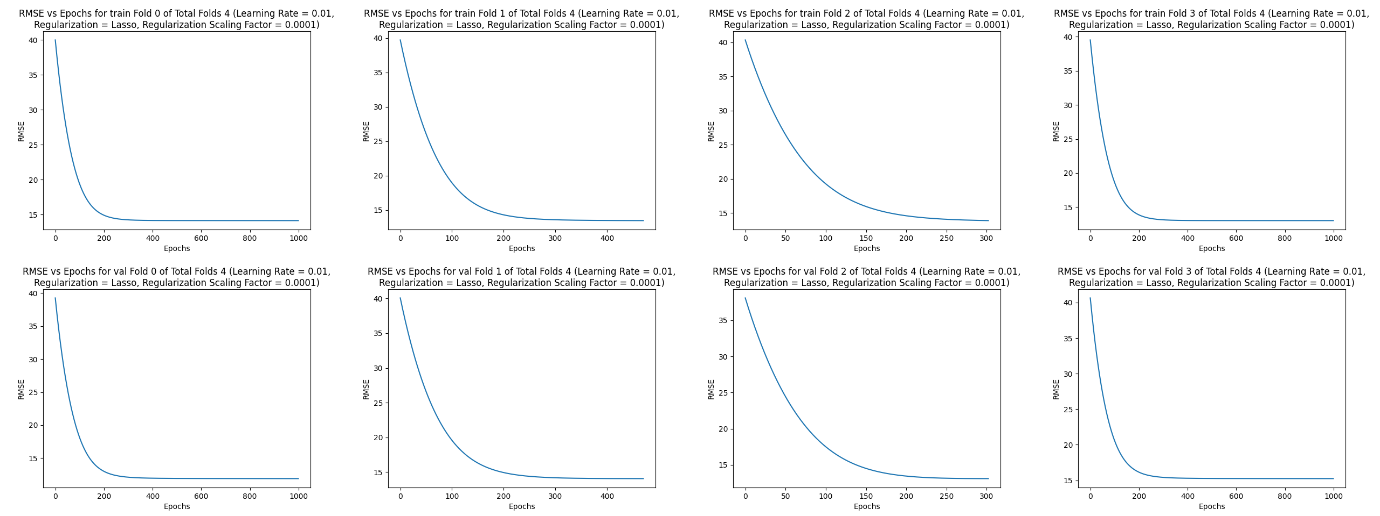


Regularization Scaling Factor: 1e-4

Average Train RMSE: 13.621032434121398,

Average Validation RMSE: 13.573719066838224,

Average Testing RMSE: 15.368920438588797

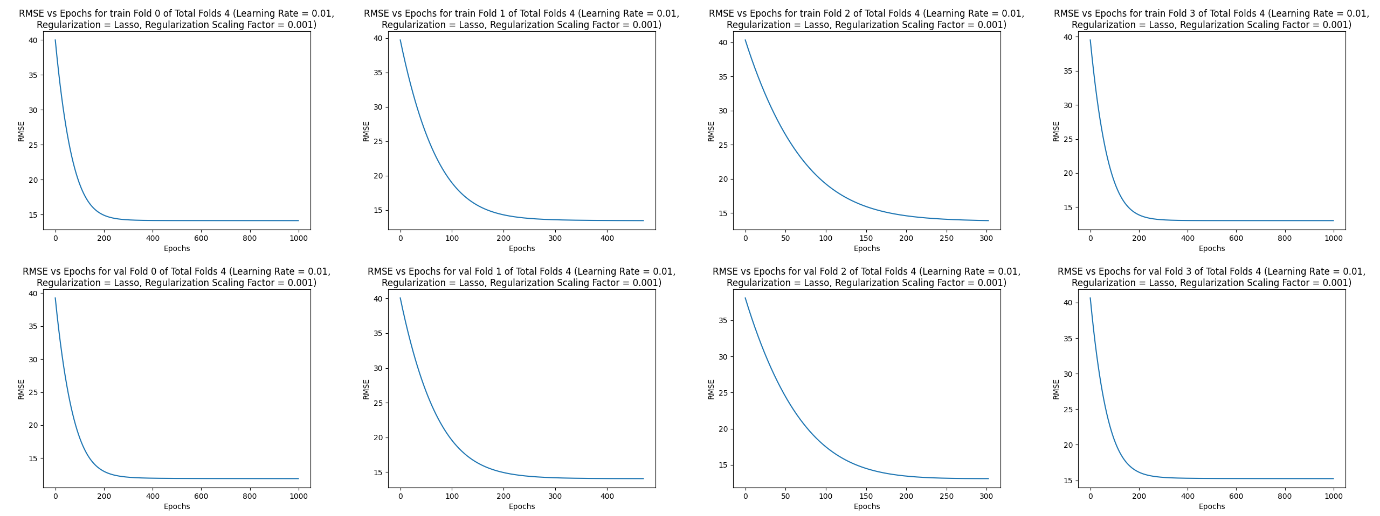


Regularization Scaling Factor: 1e-3

Average Train RMSE: 13.621032407015878,

Average Validation RMSE: 13.573718999729657,

Average Testing RMSE: 15.368920603241431

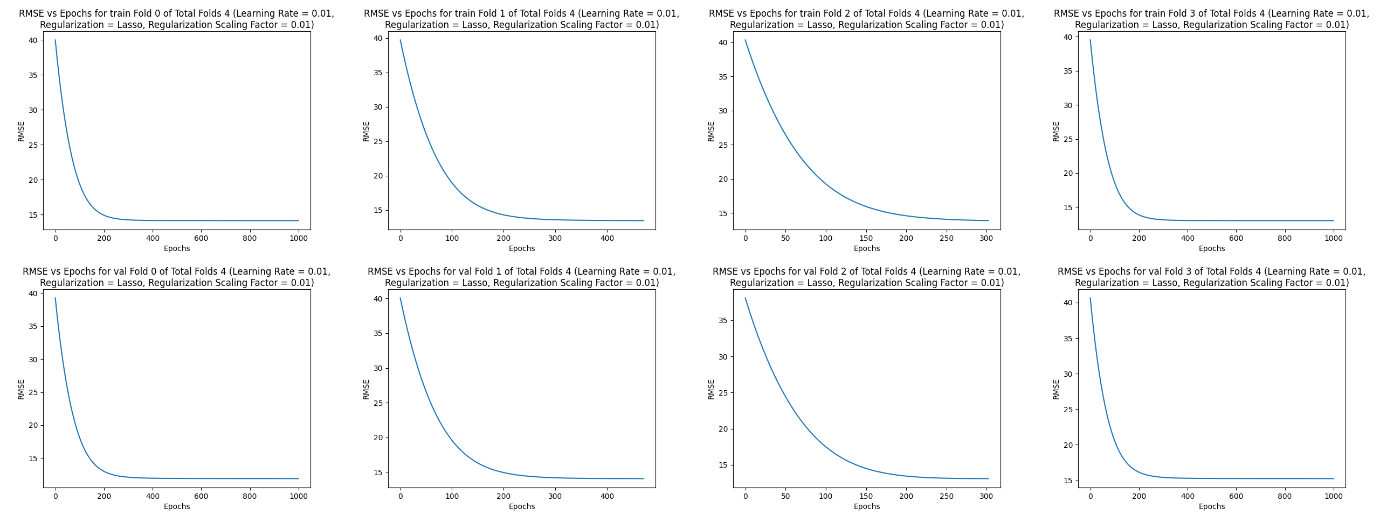


Regularization Scaling Factor: 1e-2

Average Train RMSE 13.62103213603883,

Average Validation RMSE: 13.573718328721801,

Average Testing RMSE: 15.36892224987182

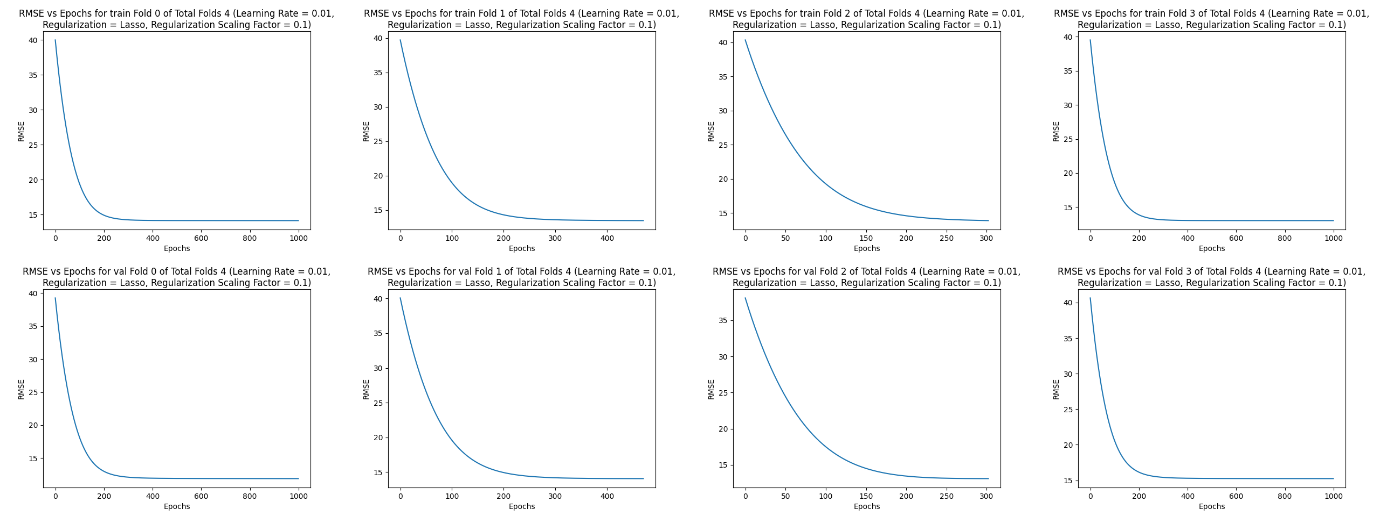


Regularization Scaling Factor: 1e-1

Average Train RMSE: 13.621029434082011,

Average Validation RMSE: 13.57371162642407,

Average Testing RMSE: 15.368938726581554

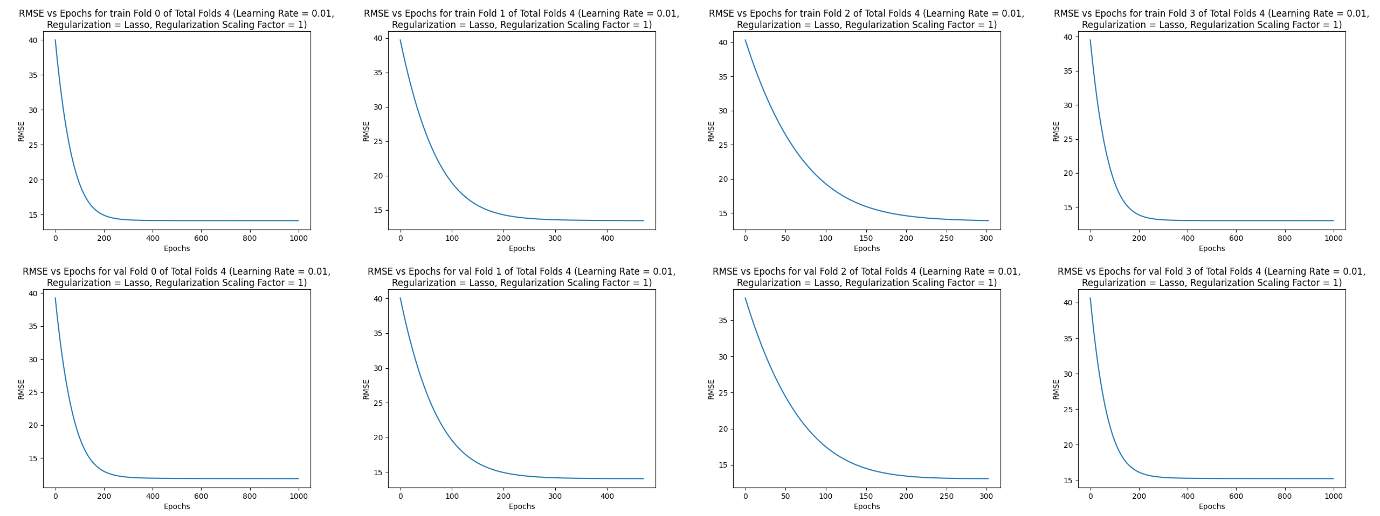


Regularization Scaling Factor: 1

Average Train RMSE: 13.62100319843991,

Average Validation RMSE: 13.57364538381816,

Average Testing RMSE: 15.36910453972574

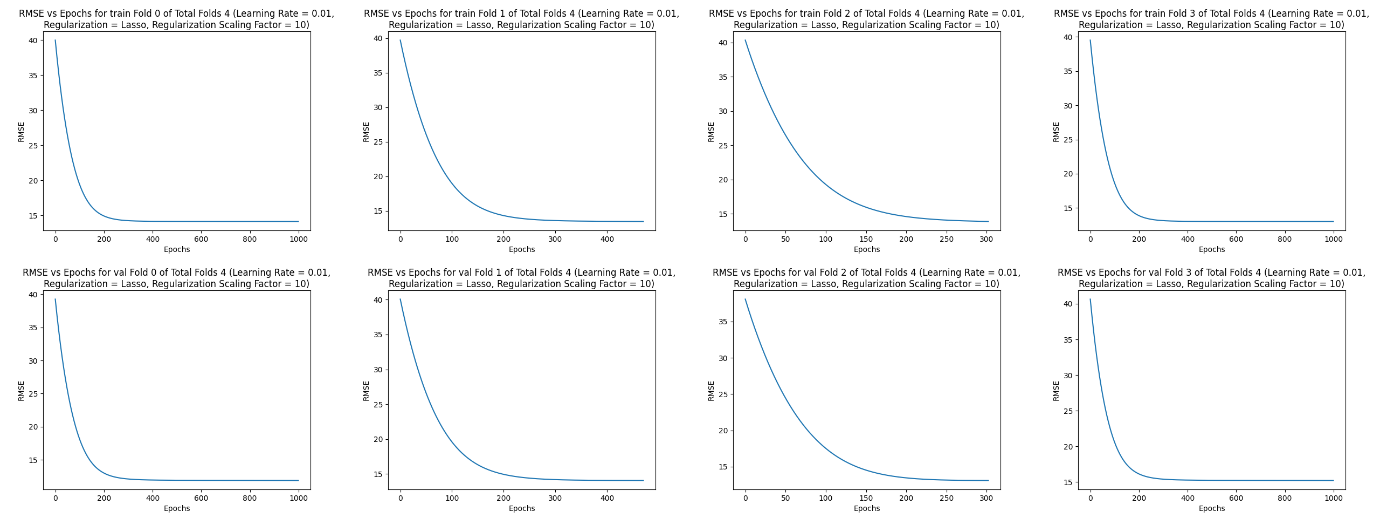


Regularization Scaling Factor: 10

Average Train RMSE: 13.620819067252006,

Average Validation RMSE: 13.57306081350232,

Average Testing RMSE: 15.37086689664329



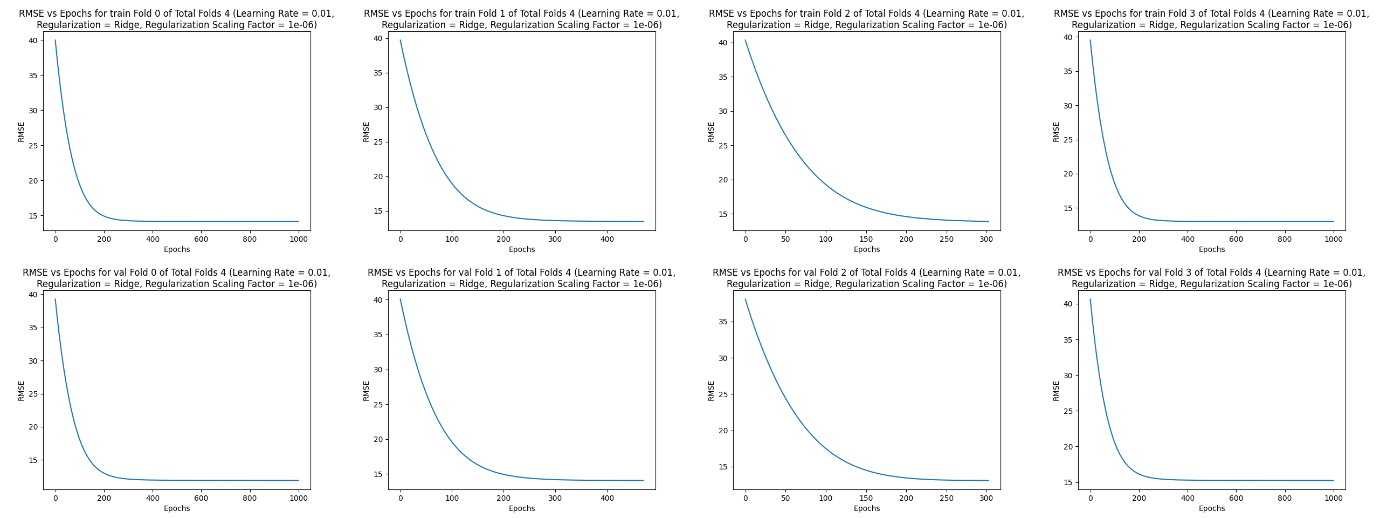
**Ridge Regression**

Regularization Scaling Factor: 1e-6

Average Train RMSE: 13.62103244465113,

Average Validation RMSE: 13.573719082756185,

Average Testing RMSE: 15.368920457223187

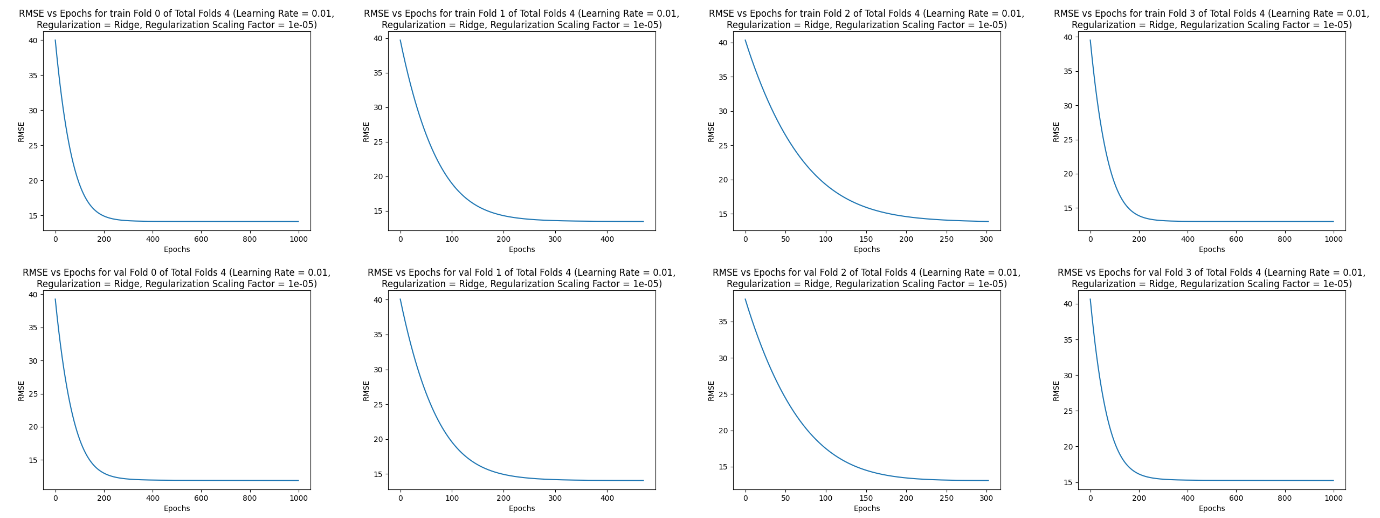


Regularization Scaling Factor: 1e-5

Average Train RMSE: 13.621032512312565,

Average Validation RMSE: 13.573719158908641,

Average Testing RMSE: 15.36892078958438

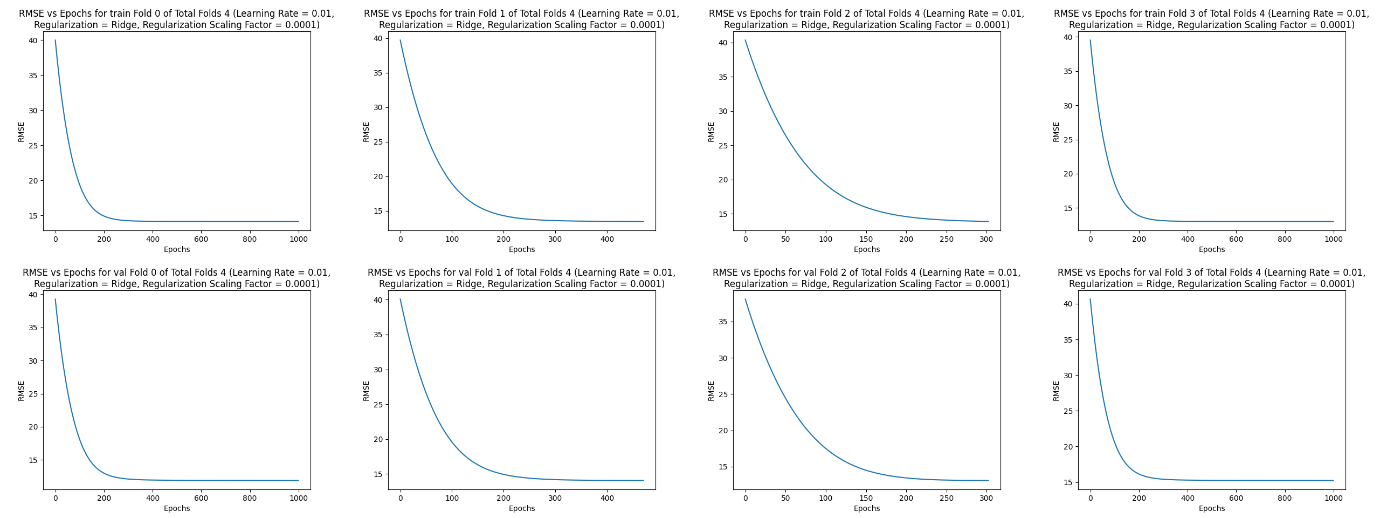


Regularization Scaling Factor: 1e-4

Average Train RMSE: 13.621033188943098,

Average Validation RMSE: 13.57371992044938,

Average Testing RMSE: 15.368924113208651

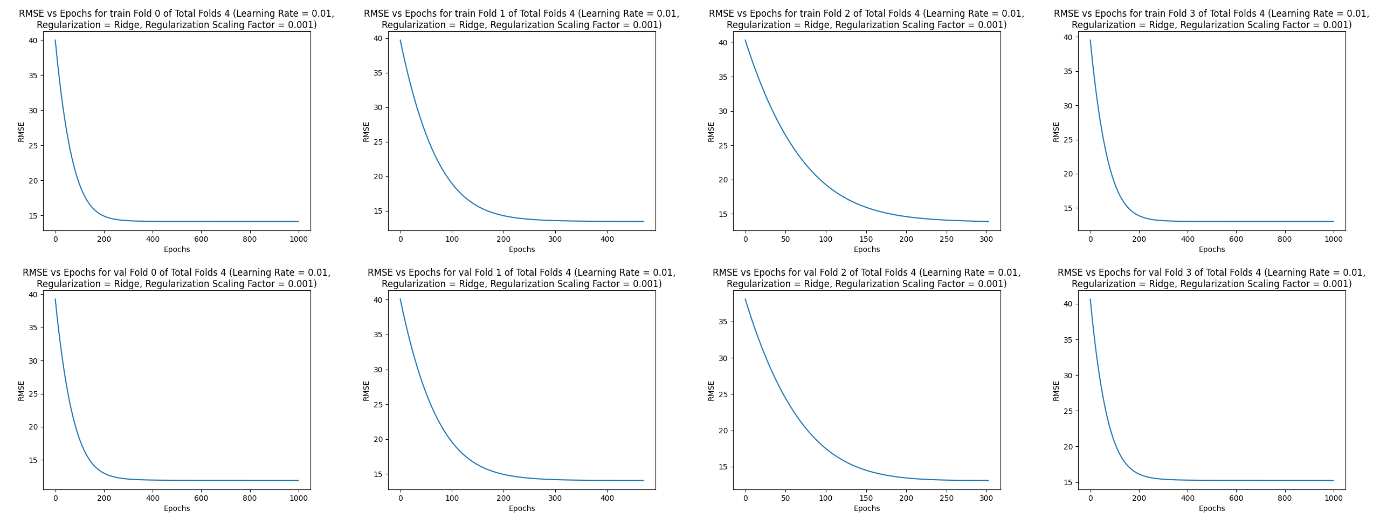


Regularization Scaling Factor: 1e-3

Average Train RMSE: 13.62103995686404,

Average Validation RMSE: 13.573727537473026,

Average Testing RMSE: 15.368957350685061

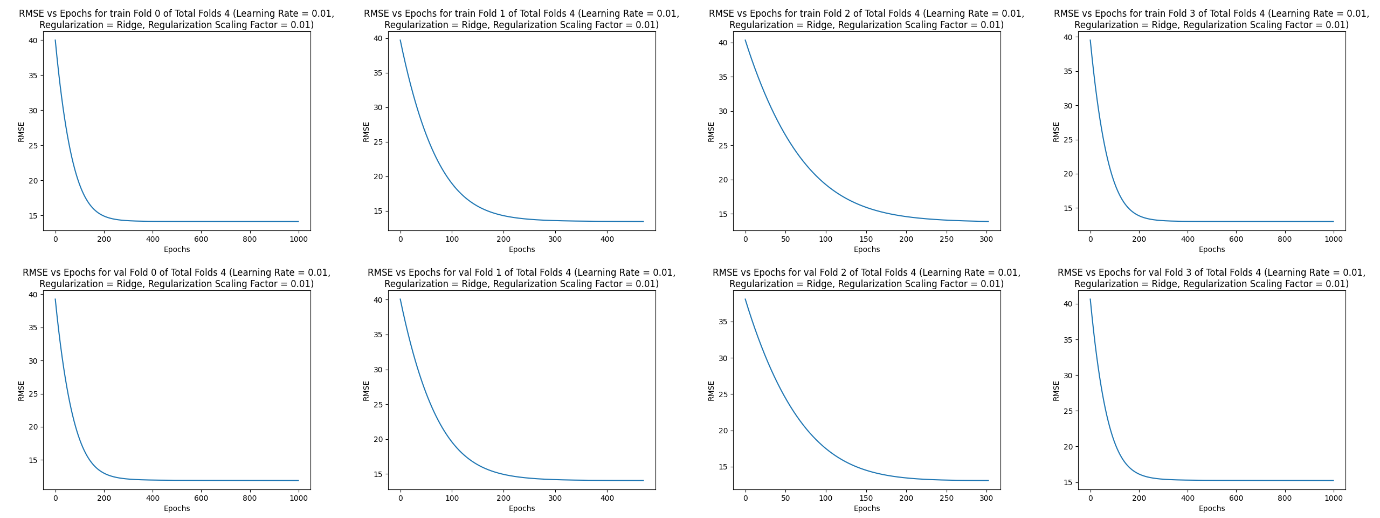


Regularization Scaling Factor: 1e-2

Average Train RMSE: 13.621107797618485,

Average Validation RMSE: 13.573803869317862,

Average Testing RMSE: 15.369289848803497

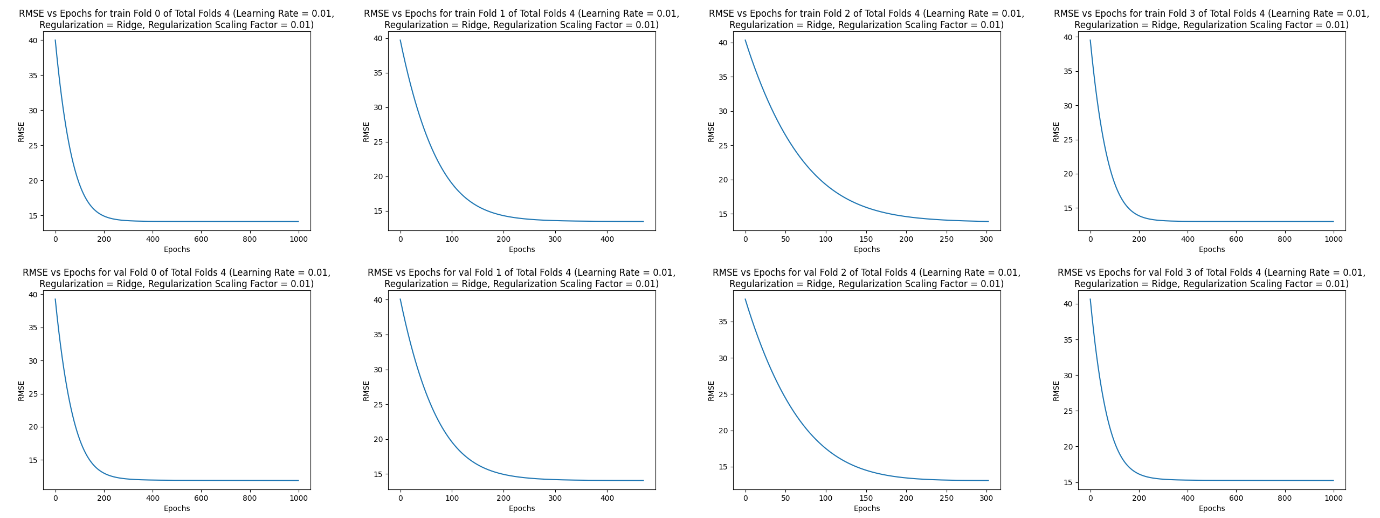


Regularization Scaling Factor: 1e-1

Average Train RMSE: 13.622372080434005,

Average Validation RMSE: 13.574580099461734,

Average Testing RMSE: 15.373660480306665

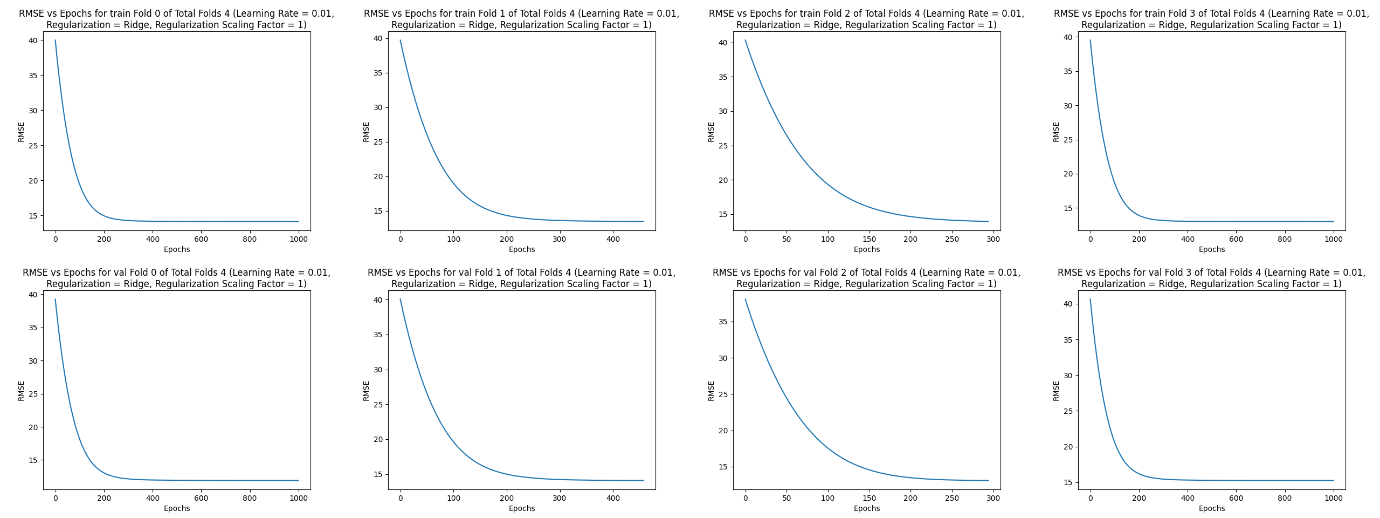


Regularization Scaling Factor: 1

Average Train RMSE: 13.635957963979989,

Average Validation RMSE: 13.584766693347426,

Average Testing RMSE: 15.417028278701533

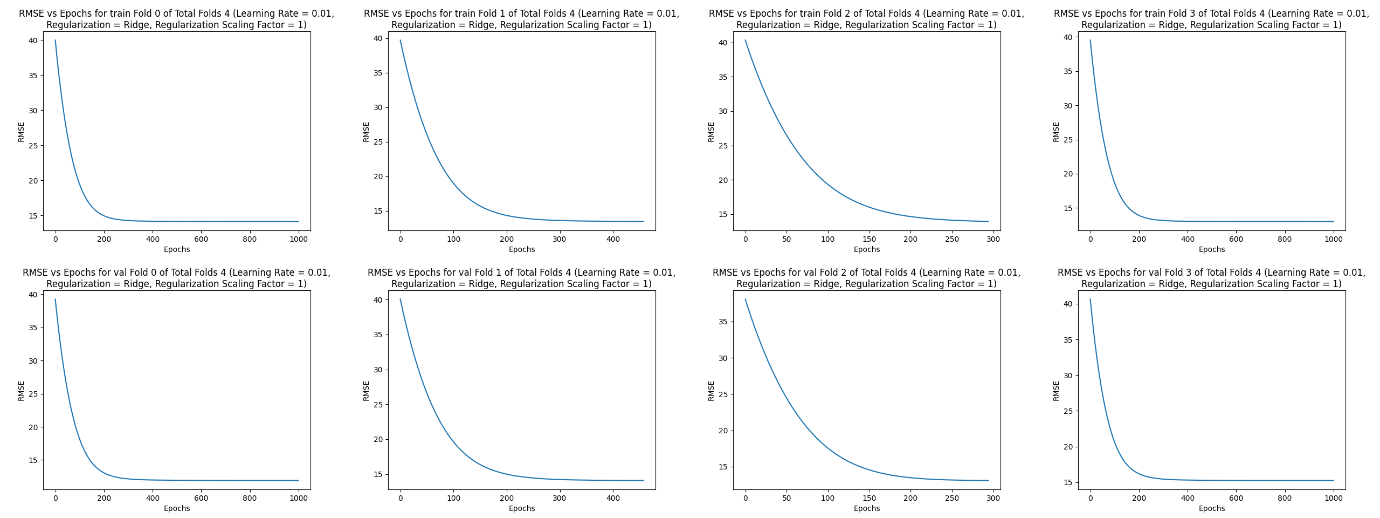


Regularization Scaling Factor: 10

Average Train RMSE: 14.016291780928125,

Average Validation RMSE: 13.951831670053664,

Average Testing RMSE: 16.075232327713646



Conclusions:

As you can see the values are really close across all regularization parameters. This can be attributed to the fact that even without regularization the model was converging and Regularization only gives small nudges to the model. Also due to early stopping the model training stops as soon as improvement halts hence the convergence point is similar! Also, even without regularization the selected configuration of best learning rate with best value of K was easily converging.

Between Ridge and Lasso Regression we can see regularization affects the final RMSE in the case of Ridge more as the model weights are small and squaring them makes them even smaller in many cases

Note: All code has been designed to make runs as reproducible as possible by using seeds wherever randomness is encountered

d)

Optimal k value is 4

Solution using Normal Form –

|  |  |  |  |
| --- | --- | --- | --- |
| Fold Number | Train RMSE | Validation RMSE | Test RMSE |
| 1 | 13.562449045863227 | 13.39722467512512 | 12.625907588423487 |
| 2 | 13.655206582285606 | 13.106350073081536 | 12.648832811116524 |
| 3 | 13.128930459904227 | 14.695714200786362 | 12.875102835895891 |
| 4 | 13.592672251401538 | 13.392851407397691 | 12.878341990659598 |

**C.**