SMAI Assignment 2

Naila Fatima
201530154

Problem 4: Least Squares Regression

a) Lasso:

Best Case:

Using sklearn's lasso regression model with alpha = 0.000001 and maximum iterations = 10000, normalize = True and tol = 0.0001, the accuracy, precision, recall and f1 scores are 0.75, 0.83471074380165289, 0.75939849624060152 and 0.79527559055118113. The threshold used was *0.568* and this is the best performance.

Keeping alpha constant and decreasing the threshold to 0.52, the metrics become 0.721153846154, 0.76978417266187049, 0.80451127819548873 and 0.78676470588235303. We can see that there is a decrease in performance for all metrics except recall which shows an increase. On increasing the threshold to 0.585, the metrics become 0.711538461538, 0.82300884955752207, 0.6992481203007519 and 0.75609756097560976.We can see that all the performance metrics have decreased in this case when compared to having a 0.568 threshold.

Case 1:

On increasing alpha to 0.1 and keeping the threshold 0.568, we note that the metrics become 0.639423076923, 0.63942307692307687, 1.0 and 0.78005865102639294. This shows a clear decrease in all the metrics except recall which is now 100%. On keeping alpha constant but decreasing threshold to 0.52, the metrics show no change at all. On increasing threshold to 0.568, there is still no change.

I have noticed that on increasing the alpha value, the metrics remain constant irrespective of the threshold provided.

Case 2:

On decreasing alpha to 0.0001 and keeping the threshold at 0.568, the metrics become 0.725961538462, 0.81666666666666665, 0.73684210526315785 and 0.77470355731225304. We can see that the metrics are better for the same threshold but alpha = 0.1. On decreasing the threshold to 0.52, the metrics become 0.706730769231, 0.76086956521739135, 0.78947368421052633 and 0.77490774907749072. We can see that while accuracy and precision have dropped, recall and f1 scores have improved. On increasing the threshold to 0.585, the metrics become 0.711538461538, 0.82300884955752207, 0.6992481203007519 and 0.75609756097560976. We can see that all have decreased except precision.

Case 3:

On decreasing alpha further to 0.00000005 and keeping threshold 0.568, the metrics become 0.75, 0.83471074380165289, 0.75939849624060152 and 0.79527559055118113. We can see that these are same as the best case. The metrics are 0.721153846154, 0.76978417266187049, 0.80451127819548873 and 0.78676470588235303 for threshold 0.52 (same as best case). Even on increasing threshold to 0.585, the metrics are same as those for best case.

Case 4:

On using same alpha as best case and threshold = 0.568 and setting Normalize parameter to false, the metrics become 0.745192307692, 0.82786885245901642, 0.75939849624060152 and 0.79215686274509811. We can note that there is a slight decrease in performance when compared to the best case but this beats most of the other cases we have seen.

Observations:

The best case is obtained by using alpha = 0.000001, threshold = 0.568 and Normalize = True. Also, we saw that on increasing and decreasing alpha beyond a certain extent, there would be no change at all in the parametrics.

b) Ridge

Best Case:

Using sklearn's ridge regression, the best case comes for alpha = 0.1, max_iter = 10000, Normalize = True and tol = 0.001 with threshold = 0.425 gives the best performance with accuracy, precision, recall and f1 scores being 0.730769230769, 0.73913043478260865, 0.89473684210526316 and 0.80952380952380953, respectively.

On reducing threshold to 0.4 (other things remaining constant), the metrics become 0.716346153846, 0.71511627906976749, 0.92481203007518797 and 0.80655737704918029. We can see that all the metrics except recall (which has increased) have decreased.

On increasing the threshold to 0.45, the metrics became 0.721153846154, 0.73885350318471332, 0.8721804511278195 and 0.79999999999999982. When compared to the best case, the performance has clearly dropped but the precision and accuracy are better than that for threshold = 0.4 whereas recall and f1 scores are poorer.

Case 1:

On decreasing alpha to 0.001 and having the threshold 0.425, the metrics become 0.716346153846, 0.73124999999999996, 0.87969924812030076 and 0.79863481228668942. We can see that the performance has clearly decreased when compared to the best case.

On keeping the threshold 0.40, the metrics decrease to 0.697115384615, 0.71084337349397586, 0.88721804511278191 and 0.78929765886287617. The performance has worsened. On keeping the threshold 0.45, the metrics become 0.730769230769, 0.74522292993630568, 0.87969924812030076 and 0.80689655172413788. The performance is clearly the best if we increase threshold for lesser alpha.

Case 2:

On decreasing the alpha further to 0.00001 and having the threshold 0.425, the metrics become 0.711538461538, 0.72670807453416153, 0.87969924812030076 and 0.79591836734693888. On changing the threshold to 0.40, the metrics become 0.697115384615, 0.71084337349397586, 0.88721804511278191 and 0.78929765886287617. The performance has degraded slightly (except for recall). On increasing the threshold to 0.45, the metrics become 0.730769230769, 0.74522292993630568, 0.87969924812030076 and 0.80689655172413788. The performance is again shown to be better if we increase threshold for smaller alpha (other things remaining same).

Case 3:

On increasing alpha to 0.5 and having threshold 0.425, the metrics become 0.6875, 0.68888888888888888, 0.93233082706766912 and 0.79233226837060711. The overall performance (except for recall) has decreased. On decreasing threshold to 0.40, the metrics become 0.6875, 0.68279569892473113, 0.95488721804511278 and 0.79623824451410652. The precision, recall and f1 scores have increased slightly. On increasing threshold to 0.45, the metrics become 0.697115384615, 0.69886363636363635, 0.92481203007518797 and 0.79611650485436891 whereall the precision and accuracy have improved but recall and f1 scores have decreased.

Case 4:

On increasing alpha to 1.0 and having threshold 0.425, the metrics become 0.668269230769 ,0.66666666666666663, 0.96240601503759393 and 0.78769230769230769. The performance is clearly reducing when opposed to the case of decreasing alpha. For threshold 0.4, the metrics become 0.649038461538, 0.65000000000000002, 0.97744360902255634 and 0.78078078078078084. We can see that all but recall have decreased. For threshold 0.45, the metrics become 0.682692307692, 0.68108108108108112, 0.94736842105263153 and 0.79245283018867918 showing that the performance improves in all scores except for recall (where it decreases).

Case 5:

Using the alpha = 0.1 and threshold = 0.425 (same as best case), but making Normalize = False, the metrics become 0.730769230769, 0.74522292993630568, 0.87969924812030076 and 0.80689655172413788. This shows a slight decrease in performance.

Observations:

On increasing alpha beyond an extent, the performance degrades. Also, for smaller alpha, the performance improves for larger threshold. Similarly, for larger alpha, the performance tends to improve for increasing thresholds.

c) Elastic Net

Best Case:

For sklearn's elastic net regression model, the best case occurs for alpha = 0.00006, Normalize = True, max_iter = 10000 and threshold = 0.48. The accuracy, precision, recall and f1 scores are 0.745192307692, 0.77027027027027029, 0.8571428571428571 and 0.81138790035587194, respectively.

On reducing threshold to 0.46, the metrics become 0.730769230769, 0.74522292993630568, 0.87969924812030076 and 0.80689655172413788. We can see that all the metrics except recall (which shows an improvement) have deceased.

On increasing the threshold to 0.5, the metrics become 0.716346153846, 0.76056338028169013, 0.81203007518796988 and 0.78545454545454541. We can see that all the metrics are clearly lower when compared to the threshold = 0.48 case.

Case 1:

On increasing alpha to 0.0001 and keeping the other parameters constant (threshold = 0.48), the metrics become 0.740384615385, 0.76870748299319724, 0.84962406015037595 and 0.80714285714285716. All of them have reduced very marginally.

On keeping threshold at 0.46, the metrics become 0.730769230769, 0.74838709677419357, 0.8721804511278195 and 0.80555555555555558. We can see that while all other metrics have reduced, recall has improved (is even better than the best case).

On increasing threshold to 0.5, the metrics become 0.721153846154, 0.76223776223776218, 0.81954887218045114 and 0.78985507246376818. All are worse (except precision) than what was obtained by decreasing the threshold.

Case 2:

On increasing alpha to 0.001 and keeping threshold at 0.48, the metrics become 0.725961538462, 0.74358974358974361, 0.8721804511278195 and 0.80276816608996537. On decreasing the threshold to 0.46, the metrics become 0.730769230769, 0.73619631901840488, 0.90225563909774431 and 0.81081081081081074. We can see that all of them show an improvement except for precision. On increasing the threshold to 0.5, the metrics become 0.725961538462, 0.75, 0.8571428571428571 and 0.79999999999999993. We can see that except for precision, the other metrics have reduced. From case 1 and 2, we can see that the performance increases by reducing the threshold.

Case 3:

On reducing alpha to 0.00001 and keeping threshold at 0.48, the metrics become 0.740384615385, 0.7651006711409396, 0.8571428571428571 and 0.80851063829787229. On reducing the threshold to 0.46, the metrics become 0.730769230769, 0.74522292993630568, 0.87969924812030076 and 0.80689655172413788. The performance has clearly reduced. On increasing threshold to 0.5, the metrics become 0.725961538462, 0.76388888888888884, 0.82706766917293228 and 0.79422382671480141. We can see that all of them have reduced except for precision.

Case 4:

On reducing alpha to 0.000001 and keeping threshold at 0.48, the metrics become 0.735576923077, 0.76000000000000001, 0.857142857142857 and 0.80565371024734977. On reducing the threshold to 0.46, the metrics become 0.735576923077, 0.75, 0.87969924812030076

and 0.80968858131487886. We can see that the recall and f1 scores have improved whereas the precision has decreased and accuracy is same. On increasing threshold to 0.5, the metrics become 0.725961538462, 0.76388888888888884, 0.82706766917293228 and 0.79422382671480141. All have shown a slight decrease in performance except precision which has improved. We can see from case 3 and 4, that on increasing alpha, a decreasing threshold tends to work better.

Case 5:

On keeping alpha = 0.00006 and threshold = 0.48 (same as best case) but making Normalize = False, the metrics become 0.725961538462, 0.76027397260273977, 0.83458646616541354 and 0.79569892473118276. We can clearly see that there is a slight decrease in performance.

Observations:

We can see that for a given alpha, the performance tends to increase on decreasing the threshold. Also, a lower alpha generally tends to give better performance.

d) No regularization

Best Case:

Using sklearn's model, the best case is obtained for normalize = False and threshold = 0.56. The accuracy, precision, recall and f1 scores are 0.730769230769, 0.80800000000000005, 0.75939849624060152 and 0.78294573643410847, respectively.

Case 1:

On increasing the threshold to 0.58, the metrics become 0.725961538462, 0.82758620689655171, 0.72180451127819545 and 0.77108433734939752. We can see that all but precision have decreased.

Case 2:

On reducing the threshold to 0.54, the metrics become 0.725961538462, 0.79230769230769227, 0.77443609022556392 and 0.78326996197718635. We can see that on reducing the threshold, the recall and f1 scores have increased.

Case 3:

On using normalize = True and threshold = 0.56, we see no change in the metrics.

Observations:

We can see that the performance seems to be a bit better for a lower threshold as opposed to a higher one.