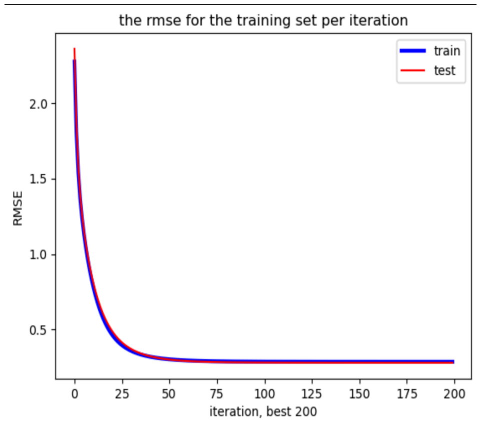
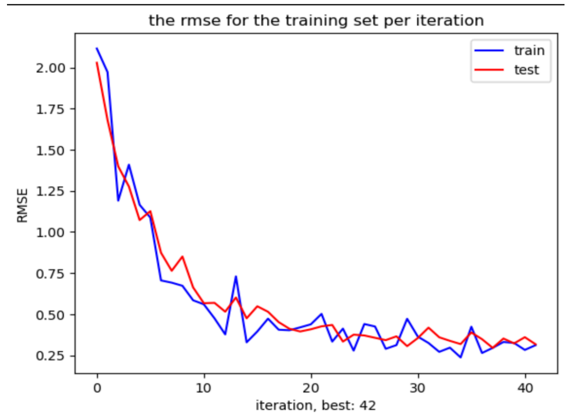
AMBROSETTI WILLIAM MIDTERM PROJECT REPORT

It is important to encode the features to be able to extract the “information” from the dataset, therefore after encoding categorical, binary and ordinal values I can apply scaling to increase the efficiency and performance for my models.

Analyzing the learning rates for the batch GD and mini batch GD I get this:

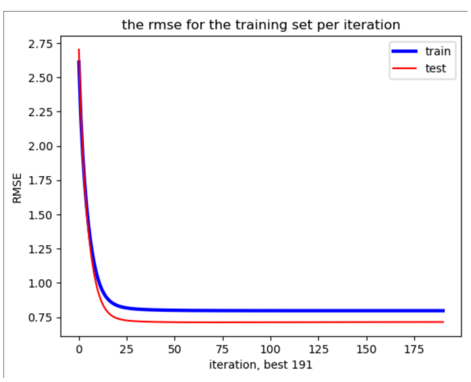


reduced data, batch GD with BMR

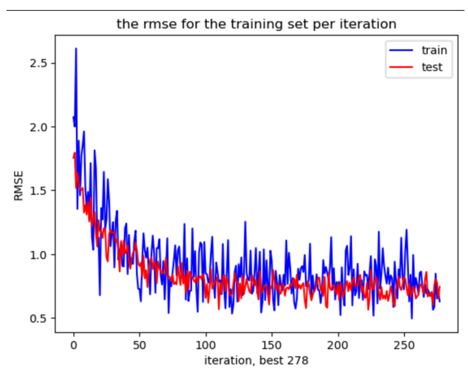


reduced data, mini batch GD with BMR

From these graphs (above and below) it is clear that the model is learning due to the Root Mean Square Error (RMSE) of the test data descending with the increase in iterations.

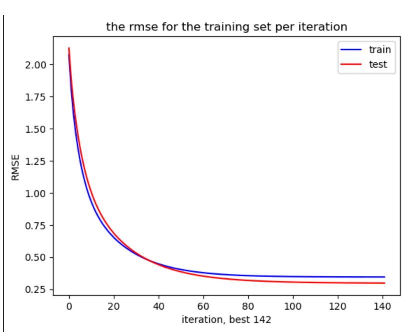


reduced data, batch GD without BMR



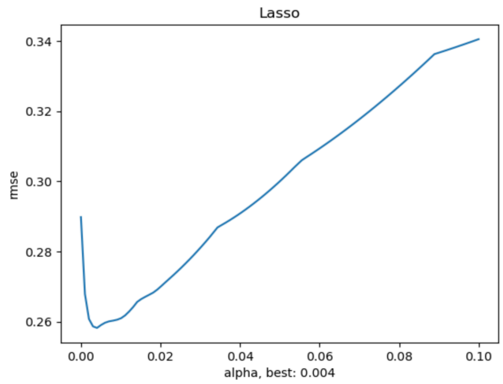
reduced data, mini batch GD without BMR

Looking at the learning rates of the data without the BMR column, it is observed that yes, the model is generalizing, but it seems that with this particular initial random value of theta the testing RMSEs are actually better in this case. As of normally this shouldn't happen, because the model isn’t learning based on the testing data but on the training data, thus the testing RMSE curve should generally be above the training one. Overall I wanted to point out the number of iterations of the mini batch GD. The reason why the number is so high is because it takes the GD without the BMR column many more iterations to reach the wanted tolerance compared to the mini batch GD done with the data having the BMR column.

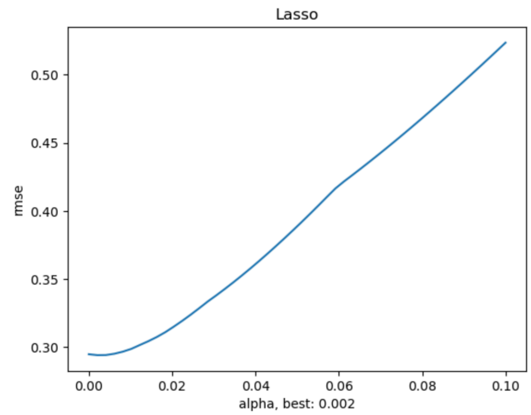


batch GD of augmented data without BMR

Considering the lateral learning curve (*batch GD of augmented data without BMR*, and the above (*reduced data, batch GD with BMR*), we can notice that the slope of the learning curve of the augmented data (especially at the start) is much more gentle (takes longer to converge) than the learning curve of the data reduced. Thus the simple removal of the BMR column (and other less important features) shows a significant impact on the learning rates and overall accuracy of the models.

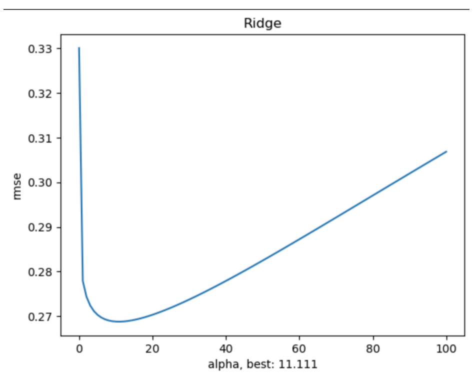


*grid search for best alpha value in augmented dataset **with BMR** with Lasso model*



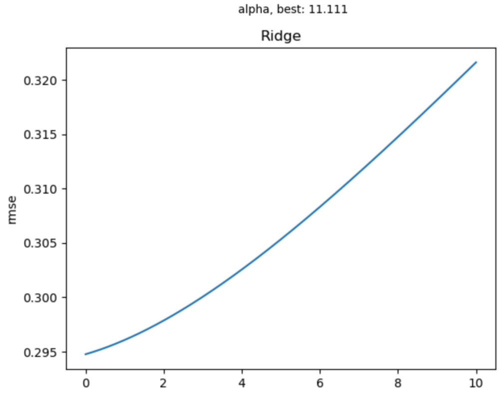
*grid search for best alpha value in augmented dataset **without BMR** with Lasso model*

(in relation to the two plots above) With a small value of alpha (removal of a few features) the augmented data performs better when having BMR in the features. But this changes when we don't have the BMR, the alpha is much smaller, meaning that to keep the RMSE low we need to keep more features than the previous plot (*grid search for best alpha value in augmented dataset **with BMR** with Lasso model)*. In addition, the score of the Lasso regression model with the BMR feature is 0.8638, but the same model without the BMR feature performs with a lower score; 0.8232.



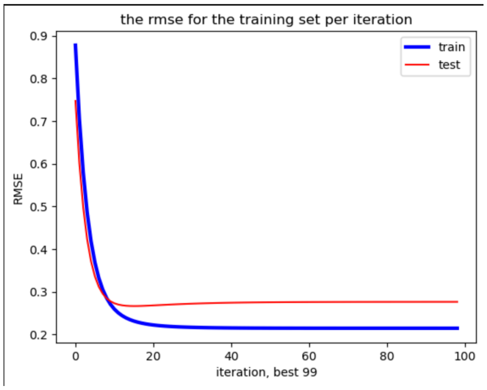
grid search for best alpha of augmented data with BMR with Ridge model

Once again the ridge regression reduces the weight of a few variables to increase the RMSE of the model (model score: 0.852).

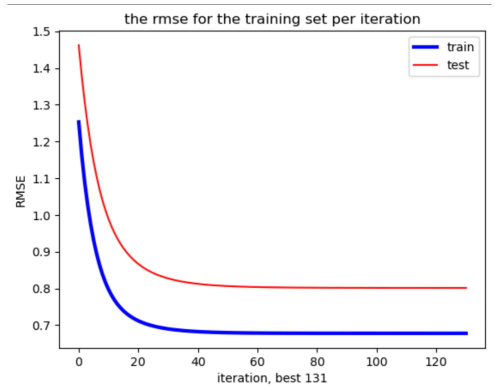


grid search for bets alpha value in augmented dataset with BMR with Ridge model

Here we can see that the model "proposes" to not lower the weight of the features to allow to reach a low enough RMSE. (model score: 0.822)



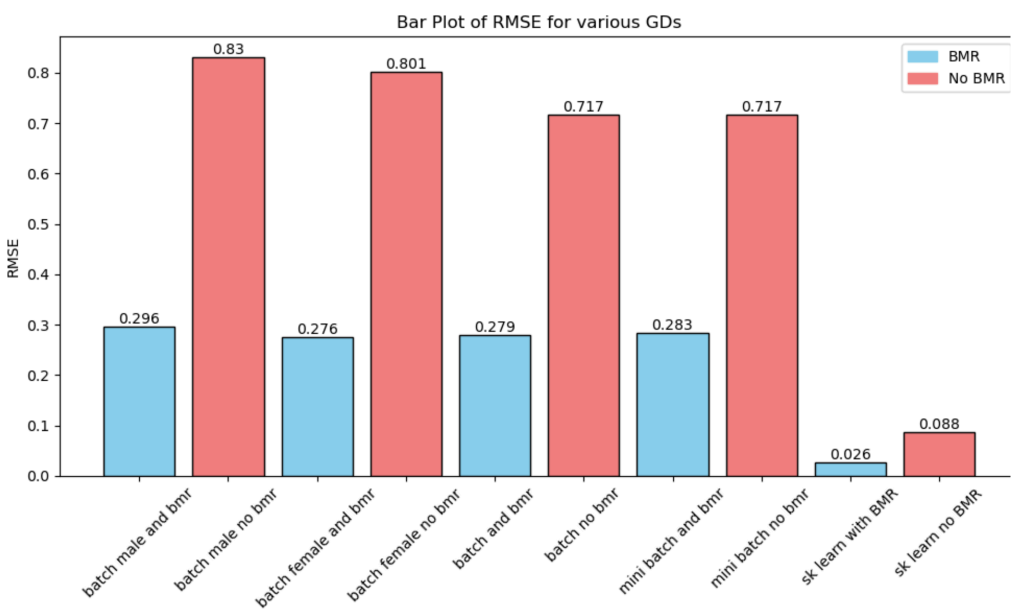
batch GD for only females dataset with the BMR column



batch GD for only females dataset without BMR

Without looking at the learning curves of the male dataset due to it being not as relevant, it can be noticed that learning curves for the female datasets, with and without BMR, show very different results. The learning curve with the BMR feature seems to have lower RMSE than the one without the BMR. Also the curve of the female data with BMR follows much closer (at the beginning) the training curve than the dataset without the BMR. With a lower amount of data (only female dataset), the model can learn the training data much better, but with the cost of not generalizing as good.

The BMR feature in the dataset is one of the most important factors to take into consideration when predicting the final weight of a person after doing the diet. This is so for all the various types of gradient descents and linear models.



Analyzing in more detail, if we take in consideration only a certain gender, a dataset with only females would perform best. This same model (females only dataset), according to this iteration of splitted data, performs as well as the global model does using batch Gradient Descent (GD) or mini batch GD. In general after seeing various runs of the notebook I always notice that the female dataset outperforms the male one, but not necessarily the global models. (general info.: I used random state 42 for all data splits)