Task 3

Student: Arno Dunstatter

EID: 1926976

Methods

This task utilized the sklearn libraries on SVMs and MLPs. First the data was downloaded from the [UCI repository for machine learning]. The data was imported into a pandas grid and a training:testing split was taken with a 0.69:0.31 ratio. Grid search was then manually implemented with nested for-loops to tune various hyperparameters for each model-type, scoring each model-configuration on both accuracy and mean absolute error (MAE), as evaluated by 10-fold cross validation without shuffling (*validation_metric* below). After each model was evaluated with 10-fold cross validation its hyperparameters and performance metrics were appended as a list to a list of all models, which could thereafter be easily sorted by either the accuracy or the MAE depending on which was being prioritized as a performance metric. The models which resulted in the best accuracy and the best MAE for both model-types were selected for testing, whereupon 10-fold cross validation was repeated for each model, with each having been already trained on the entirety of the training set prior to this round of testing via 10-fold cross validation. This testing round of cross validation attempted to avoid overly optimistic or pessimistic evaluations by utilizing the average over 1000 rounds of 10-fold cross validation with shuffling enabled to thereby mitigate the effects of random-chance when it comes to observation ordering, resulting in the *testing_metrics* shown below.

Results

SVM

kernel	C	gamma	class_weight	degree	validation_accuracy	testing_accuracy	validation_MAE	testing_MAE
rbf	4	scale	None	3	0.23	0.241755	-1.54	-1.992665
poly	1	auto	None	1	0.25	0.272985	-1.95	-2.166130

MLP

hidden_layer_sizes	activation	solver	learning_rate	learning_rate_init	validation_accuracy	testing_accuracy	validation_MAE	testing_MAE
(100,)	relu	sgd	constant	0.01	0.17	0.23384	-1.48	-2.111435
(50,)	relu	adam	constant	0.01	0.24	0.21119	-1.95	-2.163365

Discussion

Due to the “No Free Lunch” Theorem in machine learning it cannot be said that either SVMs or MLPs are superior over the other in general, however when it comes to this particular data set it appears from the above results that the SVMs, on average, marginally out performed the MLPs both when judged by accuracy and by MAE. One possible explanation asserts that SVMs do better at classification than MLPs because while MLPs merely try to minimize the sum-of-square errors by adjusting network weights between nodes, SVMs instead try to explicitly determine decision boundaries between classes. An article on science direct concurs, but further claims that MLPs often constrain the number of hidden layers employed therein so as to mitigate time complexity costs associated therewith, thereby preventing the implementation of hidden layers which would otherwise benefit the model-type’s prediction performance when dealing with high dimensional data sets. It’s my suspicion that the sklearn library’s implementation of MLPs likely imposes such a constraint to shorten model training times, but to the detriment of the prediction performance. While our data set is not particularly high dimensional with 31 predictor features, these number of features are not particularly low in comparison to the dataset size of 145.

The minimizing of the MAE for each model type was performed by computing the *validation_MAE* for each model and including it in the list which described each model’s hyperparameters and performance metrics. To optimize for low MAEs (high negative MAEs in the code) the list was then sorted such that low absolute values of the MAE came first and the first model for the lists of all SVM models was selected for testing and all MLP models.

Conclusion

In summary mediocre prediction performances were observed for both model-types, even after searching through hundreds of hyper-parameter configurations for each. It’s my suspicion this had to do with the somewhat high ratio between the number of predictors (31) and the total amount of observations in the dataset (145). Assuming all features are relevant, obtaining a larger dataset would likely be beneficial to improving these model-types’ prediction performances. Otherwise a different model type, such as the Radial-Basis Function Neural Network could be employed in their stead, as described in the relevant paper.