First Home Assignment

Aprendizagem Automática

2022/2023

João Fé
fc60427@alunos.fc.ul.pt
24 WHs

Duarte Raposo
f59099@alunos.fc.ul.pt
22 WHs

Daniel Cabral
fc54790@alunos.fc.ul.pt
20WHs

1 Methodology

Firstly, we try to understand our data and see if there is any evident correlation between each of the given input variables and the dependent variable(s) (`motor_UPDRS` or/and `total_UPDRS`). Unfortunately, no clear correlation can be seen when plotting the data. However, with the analysis, it is possible to see that there is a high correlation between the values in the different units for both `jitter` and `shimmer`. This indicates that we could choose one unit for each other, and ignore the others.

Secondly, we prepare the data. Since instances are originally sorted by subject, we decide to shuffle them first. Then, we split the set into a train set and an independent validation set (IVS) – 80% for the former and 20% for the latter. Finally, we normalize the data by subtracting the mean and dividing the standard deviation.

With the data prepared, we choose our validation technique – k-fold cross-validation – and implement it in a way that is usable for every model that we are going to train and test. We then define a set of models and respective hyper-parameters that we want to test. Most of this initial range of hyper-parameters uses logarithmic steps to speed up the search. After this first test, we can get an idea of where the best parameters lie and we do a second test focusing around those parameters. We do a second test and based on the selected metrics we choose our final models for regression and classification.

Finally, we test the selected models against the IVS and conclude their final performance.

2 Implementation

To validate the tested models k-fold cross-validation is used. Considering that the dataset is relatively small, we could use $k = 10$ without sacrificing too much the processing time to train and test the models. (Even $k = 20$ could be used, however eletricity is expensive.)

For the regression model the considered metrics are mean average error (MAE), ratio of the variance explained (RVE), and R_2. For the classification model the considered metrics are accuracy, F1-score, recall, and precision.

2.1 Regressor (Objective # 1)

For the regression model selection, the tested models are the *Tree Regression*, *Linear Regression*, *Ridge*, and *Lasso*. The set of tested hyper-parameters for each model are shown in Table 1. The average results obtained over the k test sets are shown in Tables 2 to 3, with Table 3a including the results for the Linear Regression (when alpha = 0).In bold we can find the best results (all in Table 2). Tree regression `(friedman_mse,best,10)` yields the best results in every considered metric. Thus, this is our selected regression model that will be tested against the IVS.

Model	Hyper-parameters		
Tree Regressor	Criterion	Splitter	Maximum depth
	{squared_error, friedman_mse}	{best}	{5,10,20, 50,Inf.}
Linear Regression			
Ridge	Alpha		
	10^i, for $i = -5, -4, \dots 1$		
Lasso	Alpha		
	10^i, for $i = -5, -4, \dots 1$		

Table 1: Regressors' tested hyper-parameters.

Criterion	Splitter	Max. depth	MAE	RVE	R_2
squared_error	best	5	4.543	0.457	0.456
		10	3.570	0.521	0.521
		20	3.622	0.503	0.502
		50	3.601	0.503	0.503
		Inf.	3.571	0.512	0.512
friedman_mse	best	5	4.534	0.460	0.459
		10	**3.539**	**0.528**	**0.527**
		20	3.582	0.510	0.510
		50	3.590	0.505	0.504
		Inf.	3.606	0.501	0.501

Table 2: Tree regression average results over k.

2.2 Classifier (Objective # 2)

For the classification model selection, the tested models are the *Tree Classification*, and *Logistic Regression*. The set of tested hyper-parameters for each model are shown in Table 4. The obtained results for the tree classification model are shown in Table 5. The average results for the logistic regression model are 0.697 of accuracy, 0.392 of F1-score, 0.575 of recall, and 0.301 of precision. The tree classification model clearly outperforms the logistic regression. The bold values in Table 5 show the best results obtained. Different models win on different metrics, however, we decide to select model (gini, best, 5) because it is smaller than model (gini, best, 50) and has better accuracy, F1-score, and recall than model (entropy, best, 5) while achieving only 1% less precision.

Alpha	MAE	RVE	R_2
0	6.470	0.107	0.106
10e-5	6.470	0.107	0.106
10e-4	6.470	0.107	0.106
10e-3	6.470	0.107	0.106
10e-2	6.470	0.107	0.106
10e-1	6.470	0.107	0.106
1	6.470	0.107	0.106
10	6.472	0.109	0.108

(a) Ridge average results over k. Note that when $\alpha = 0$, the objective is equivalent to ordinary least squares, i.e., we have Linear Regression.

Alpha	MAE	RVE	R_2
10e-5	6.470	0.107	0.106
10e-4	6.470	0.107	0.106
10e-3	6.471	0.108	0.107
10e-2	6.486	0.109	0.108
10e-1	6.696	0.074	0.073
1	6.938	0.000	-0.001
10	6.938	0.000	-0.001

(b) Lasso average results over k.

Table 3: Ridge and Lasso results for Linear Regression.

Model	Hyper-parameters		
Tree Classifier	Criterion	Splitter	Maximum depth
	{`gini`, `entropy`}	{`best`}	{5,10,20,50,`Inf.`}
Logistic Regression			

Table 4: Classifiers' tested hyper-parameters.

Criterion	Splitter	Max depth	Accuracy	F1-score	Recall	Precision
`gini`	`best`	5	**0.883**	0.610	0.535	0.734
		10	0.881	0.651	0.642	0.662
		20	0.877	0.645	0.647	0.646
		50	0.880	**0.652**	**0.653**	0.654
		Inf.	0.878	0.648	0.646	0.652
`entropy`	`best`	5	0.877	0.551	0.446	**0.746**
		10	0.878	0.636	0.619	0.657
		20	0.877	0.641	0.636	0.648
		50	0.875	0.639	0.641	0.638
		Inf.	0.876	0.636	0.630	0.644

Table 5: Tree classifier results over k.

Other combination of hyper-parameters were tested, but the results are omitted for brevity.

3 Results

Having selected the models `treereg(friedman_mse,best,10)` and `treeclass(gini,best,5)` for regression and classification respectively, we need to evaluate them against the IVS to get independent performance measurements.

For the regression model, a MAE of 3.519, a RVE of 0.555, a R2 Score of 0.550 and a Pearson Correlation Score of 0.769 (with p-value=2.53^{-230}) were obtained. Figure 1 show the residuals plot for the model.

MAE	RVE	R_2	Pearson	MCC
3.519	0.555	0.550	0.550	0.700

(a) Regression model.

ACC	F1	Recall	Precision	MCC
0.909	0.755	0.785	0.727	0.700

(b) Classification model.

Table 6: Performance on the IVS.

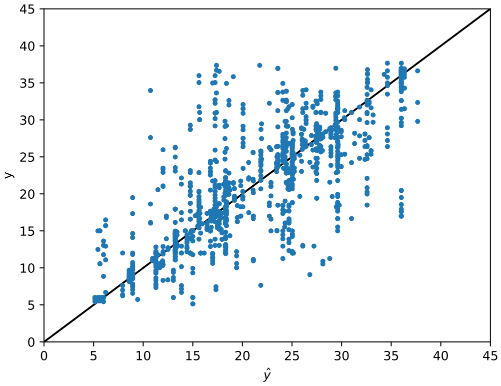

Figure 1: Residuals for the selected regression model.

For the classification model, an accuracy of 0.909, an F1-score of 0.755, a recall of 0.785, a precision of 0.727 and a MCC of 0.700 were obtained. The Table 7 represents it's respective Confusion Matrix.

		True Class	
		Negative	Positive
Predicted Class	Negative	903	62
	Positive	45	165

Table 7: Confusion matrix for classification model.

4 Discussion and Conclusion

The non-linearity of the data makes it very difficult to get a good linear regression model, that was suspected since the moment we manually analyzed the data and was confirmed when testing the regression models.

The results obtained against the IVS confirm the correctness of the developed models. The performance decrease slightly for the regression model, but that is normal because the decrease is not large. For the classification model we can even see a small performance increase when testing against the IVS – guess we were luck.