Task 2: baselines

Matthias

January 22, 2018

1 Models

I compare the following baselines in a 3 fold crossvalidation:

- a linear regression model
- an autosklearn ensemble regressor, trained for 60 seconds per fold

Both models are trained and evaluated on the raw data and preprocessed (= standardized) data. In both cases, only the configurations are used as input, not the time series (since this is task 2).

2 Expectation

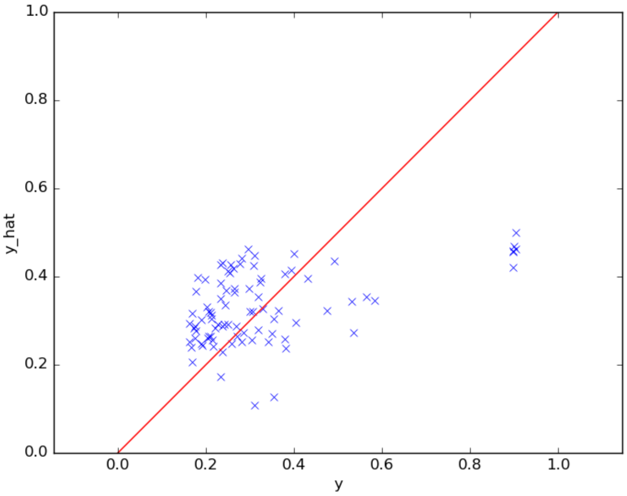
I expect the autosklearn model to be (potentially much) better than the linear regression model, since it's more powerful.

Extending the training time for the autosklearn model might improve the performance (this is not tested yet).

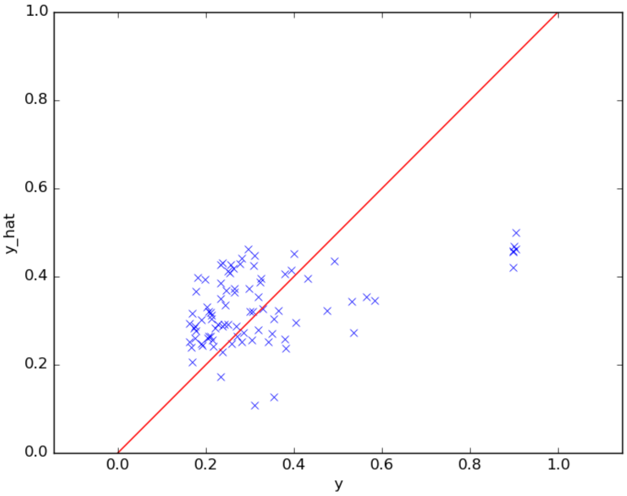
The preprocessing might improve both models' performance a little bit.

3 Results

Predicted y over true y for the linear regression model without preprocessing:

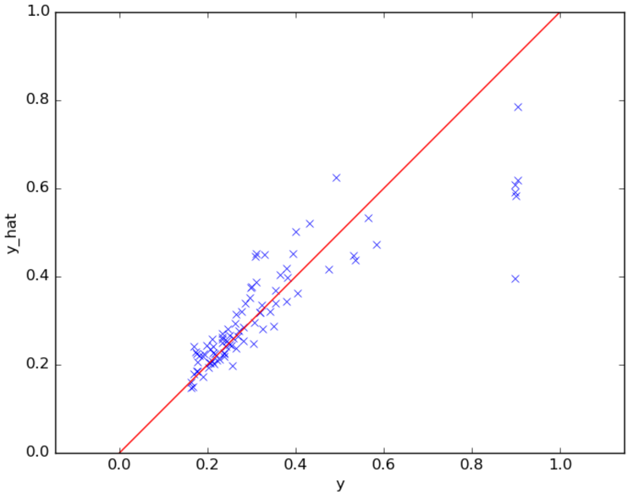


Predicted y over true y for the linear regression model with preprocessing:

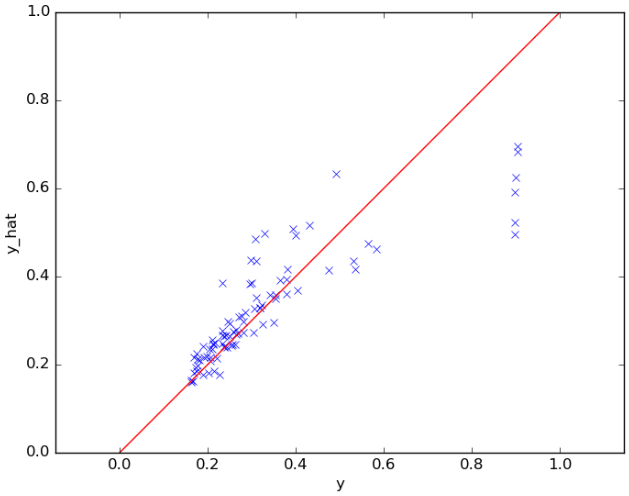


(side note: I think linear regression is invariant to scaling and shifting → that would explain why the results are the same)

Predicted y over true y for the autosklearn ensemble without preprocessing:



Predicted y over true y for the autosklearn ensemble with preprocessing:



Mean CV loss (mean squared error):

model	raw	standardized
linear regression	0.029270	0.029270
autosklearn (1 min training)	0.012119	0.010339
autosklearn (10 min training)	0.009157	xx

Keep in mind that autosklearn is randomized. I once got a mean CV loss below 0.01 for autosklearn without preprocessing with 1 min training time.
We should probably fix the random seed for autosklearn and for the k-fold splitting (I get different folds after restarting python).
Most likely autosklearn does preprocessing internally.

4 Conclusions

- preprocessing has no effect on linear regression and autosklearn
- predicting the cases with y $>$ 0.8 might be hard (it is hard for autosklearn)
- we would like to achieve a mean squared error below 0.01 to beat the baselines