Machine Learning

Implementation of LASSO Regression and Cross-validation from scratch

Milan Simovic

Output of our model is represented by hypothesis $h_\theta(x) = \hat{y} = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n$, where *x* is one example of pattern, i.e. feature vector. Therefore, we have a data set and we to determine as best as possible the dependence of the output on the input. More precisely we want to determine the parameters of θ_i, where $i = \overline{0, n}$. Training is performed according to the criteria given by expression (1).

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{i=1}^{n}|\theta_i| \tag{1}$$

Regularization is realized by adding the second part in the expression (1) that is given by $R(\theta) = \sum_{i=1}^{n}|\theta_i|$. The first addition in the expression (1) is loss function and in this case it is average sum of squared errors. LASSO regression is just one of the ways of regularization which implies that the regularization part is sum of the absolute values of hypothesis' parameters. Notice that the sum in the regularization part starts with number one which means that the regularization doesn't affect θ_0, it only penalizes parameters which multiply features. Multiple constant λ is hyperparameter that should be determined.

Optimization of this criteria is done by the gradient descent method. Values θ_i are updated, as it is showed in the expression (2), in each iteration *l*, where ρ is the learning rate.

$$\theta(l+1) = \theta(l) - \rho\frac{\partial J(\theta(l))}{\partial\theta(l)} \tag{2}$$

Gradient of this criteria is given by (3) and it is represented by vectors and matrix.

$$\nabla J(\theta) = \frac{\partial J(\theta(l))}{\partial\theta(l)} = 2X^T(X\theta - y) + \begin{bmatrix} 0 \\ sgn(\theta_1) \\ \vdots \\ sgn(\theta_n) \end{bmatrix} \tag{3}$$

In the attached code you will see class *LASSORegression* which is made so that you can type examples with any number of features and any degree of polynomial function. In this case I tried with degree of two and examples with five features from the attached data set.

That means that vector θ has 21 parameters, because in general case number of parameters is given by (4):

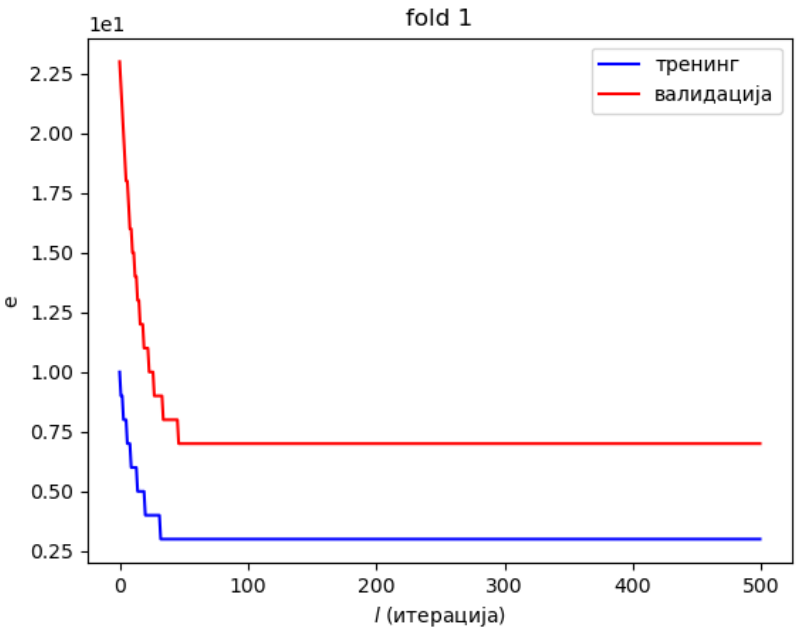
$$1 + d\binom{n}{1} + \binom{n}{2} + \binom{n}{3} + \cdots + \binom{n}{d}, \tag{4}$$

where d is degree of polynomial function and n is number of features.

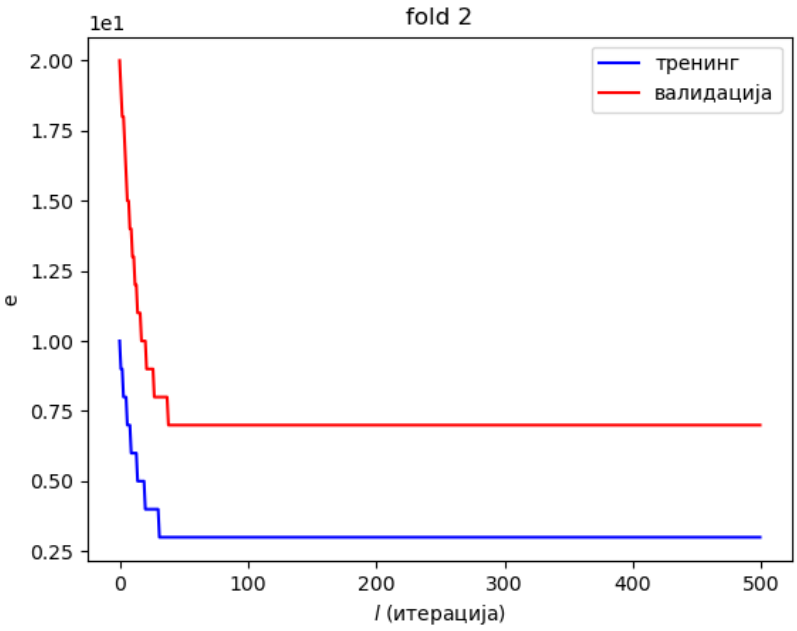
I also implemented cross-validation method. The training and test sets are split 90:10.

The training set should be divided into folds. Typical number of folds is 5 or 10. Here I've chosen five folds because of a small number of examples in this data set.

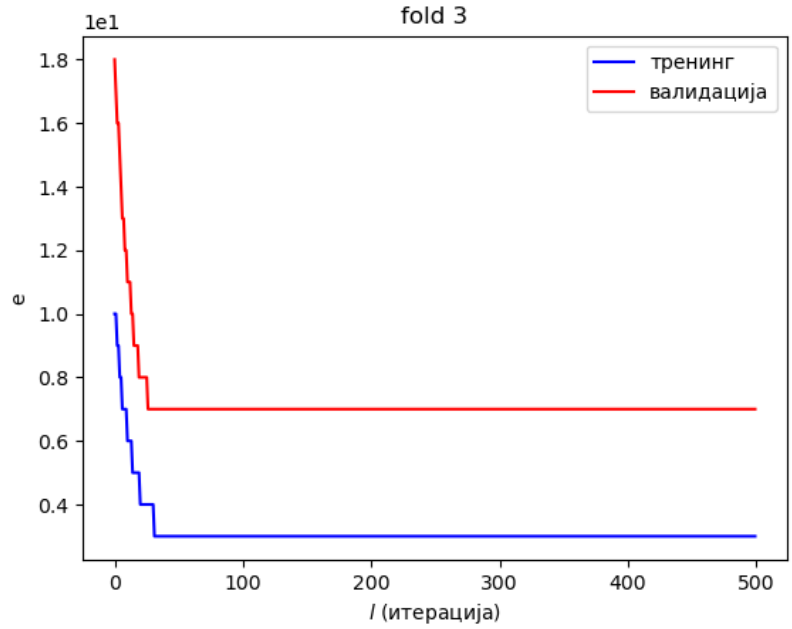
Уобичајено број тих подскупова буде пет или десет. У овом случају је због малог броја расположивих података изабрано да се изврши подела на пет подскупова. On the pictures from 1 to 5 you can see average squared error on validation and training set (in Serbian: тренинг – training, валидација – validation)



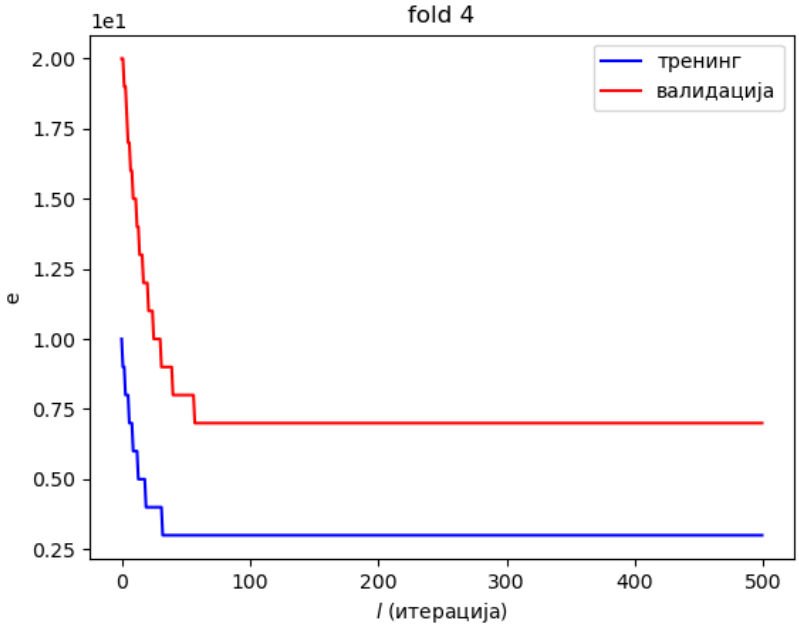
Picture 1 – error during iterations



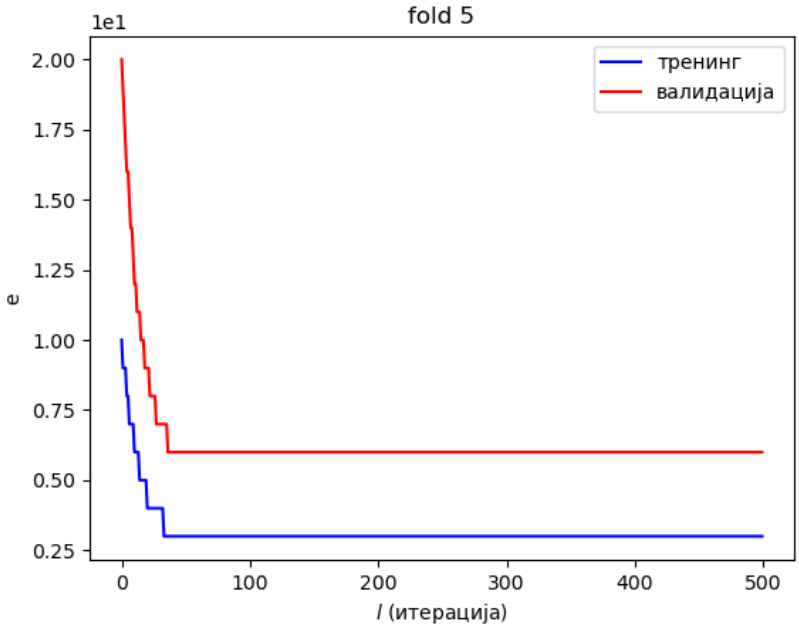
Picture 2 – error during iterations



Picture 3 – error during iterations

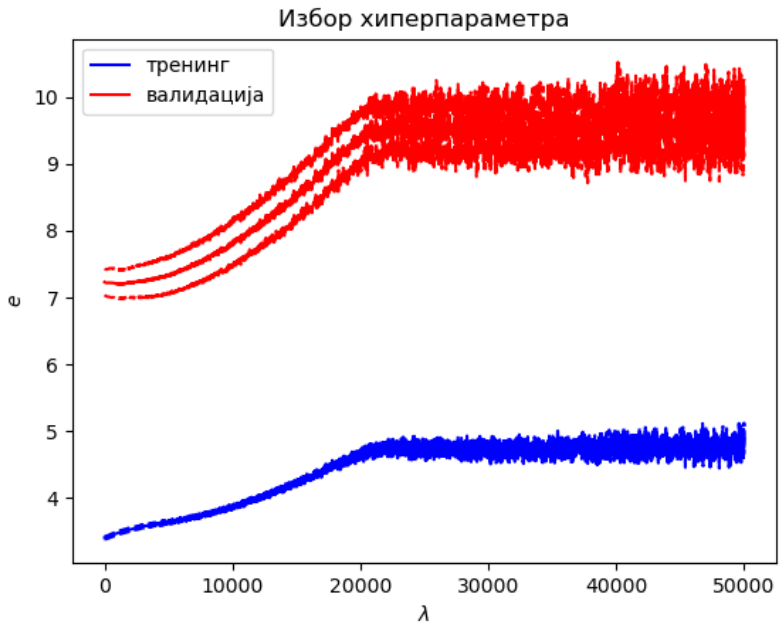


Picture 4 – error during iterations

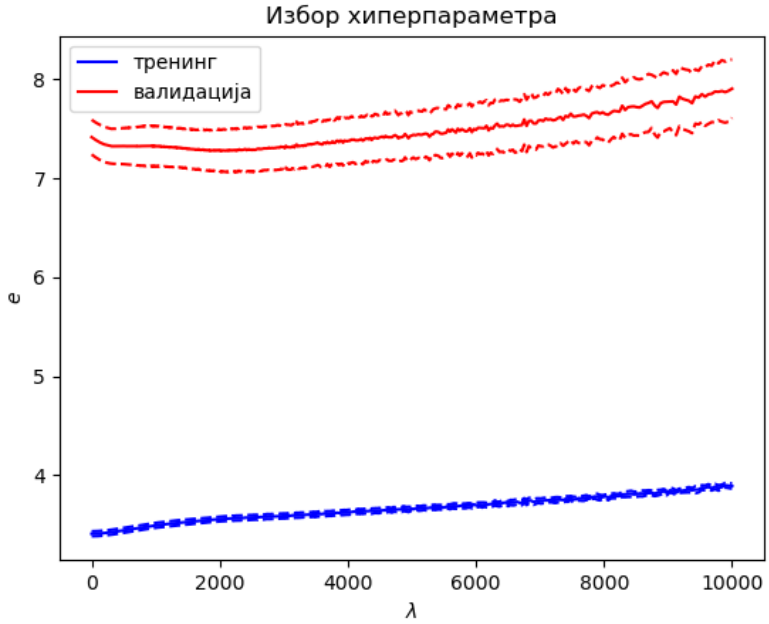


Picture 5 – error during iterations

Initially I chose wide band for λ and corresponding figure is showed on the picture 6. An appropriate figure is showed on the picture 7 (in Serbian: Избор хиперпараметра – Hyperparameter selection).



Picture 6 – too wide λ range



Picture 7 – graphic for choosing λ hyperparameter value

The solid line on this graphics represents the average value of five metrics (five average errors on five folds). And the dashed line represents the variance of that five metrics.

It is important to always show the variance of metrics because sometimes you can have more complex model with lower mean error but big variance. So it could be better if you chose simpler model with maybe a little bigger mean error but lower variance. Then you will have numerically easier model with roughly the same results

Therefore the optimal value of hyperparameter is $\lambda^* = 2026.6$. In that point the error is 7.26 ± 0.127. It is chosen on validation set of course.

Value of the root of average squared error on test set is 52.96.

Discussing the parameters of vector θ we can notice that only 4 or 5 of 21 parameters has significantly higher value than the others, and that is the consequence of implementing LASSO regression because LASSO favorizes only the most useful features.