TP10 Goupe D3_P2_B

Cazic Boris, Matthieu Ndumbi Lukuenya, Vaurs Damien

January 08, 2023

Data

Data preprocessing

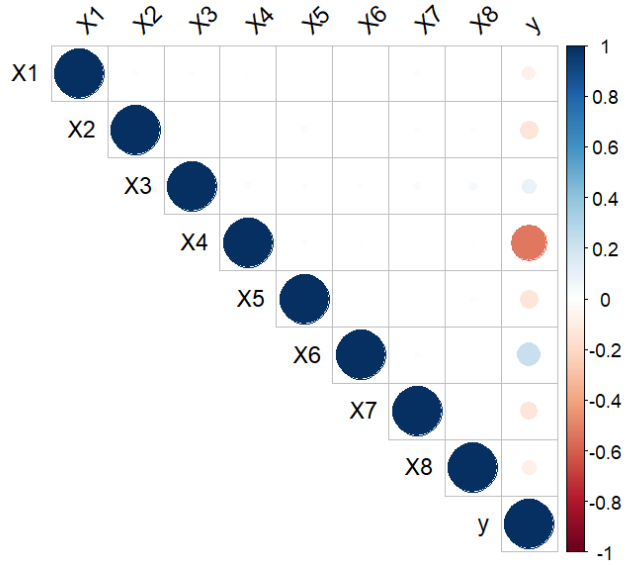
The file contains data related to kinematics of a robot arm. There are 4000 learning cases, eight predictors, and one response variable (last column).

Let's state X1,..X8 the independent variables, to represent the joints of the robot arm and Y, the dependent variable, to represent the location of the tip of the arm, given the angles of the joints.

It is a multi-dimensional regression task because we have 8 quantitative independent variables, and the problem is to predict the values of Y given X1,...X8.

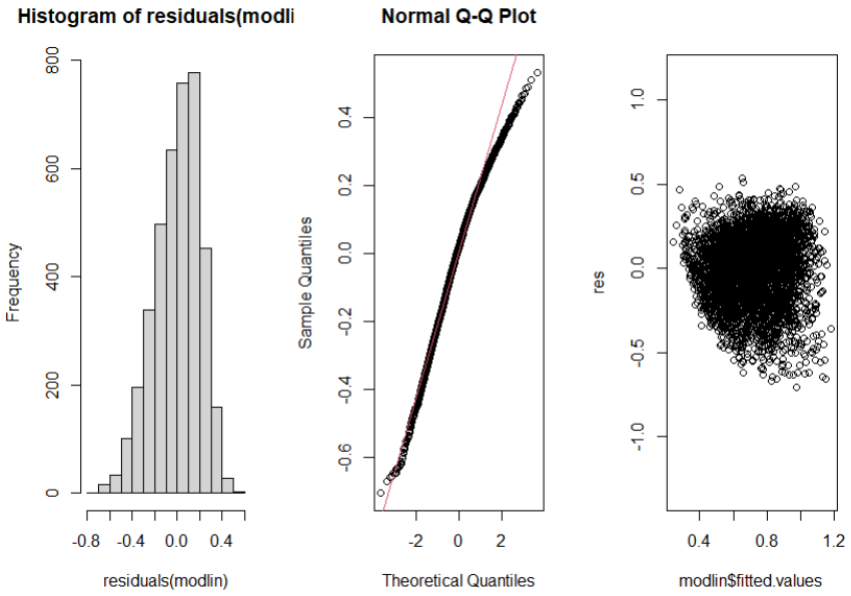
We sample the data randomly with the function sample() and partition them with a partition index of 0.8 to get 80% of the data for the training set and the rest for the test set.

Above all, let's plot our dataset in order to get a clear idea about it's distribution and the structure. For this, we make it's correlation matrix with the cor() function and plot it.



As we can see there no correlation between the variables nor with the dependent variable Y. We can also see a very weak correlation between the independent variable X4 with Y.

Model estimation and graphs of residuals



Hypotheses of linear model don't seem to be valid here because the residuals are non Gaussian (test of shapiro - wilk). However in the case of a big sample, this model remains robust and can fit. That's why we will test it.

Regularization

We will use regularization just to help the model to generalize well on the test set. Penalty selection by cross - validation

Ridge, Lasso and ElasticNet

The table represents the values of the MSE on dataset depending on the parameter alpha for the 3 models. We test these 3 different regularizations : Ridge (alpha= 0), lasso (alpha = 1) and elasticNet where the value of alpha is determined by cross validation.

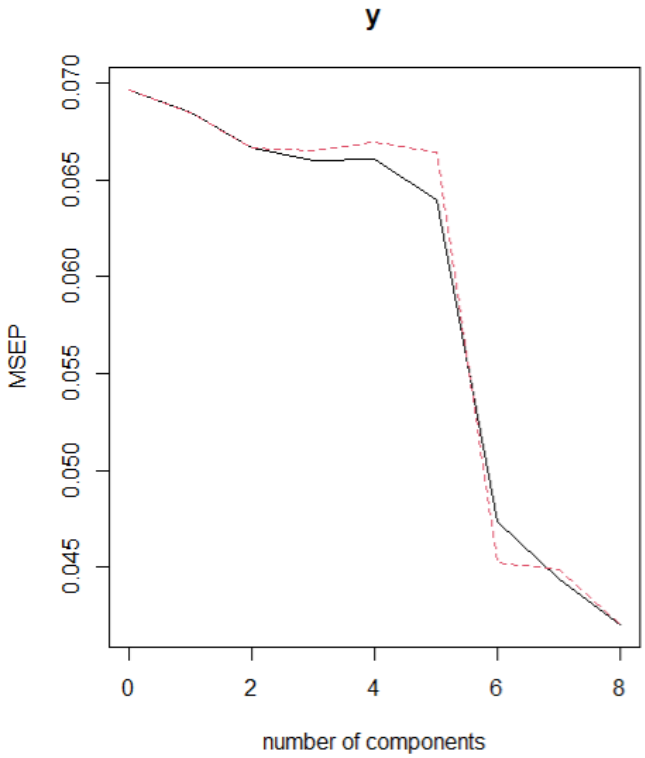
We use the glmnet package on the 3 models.

Ridge (alpha = 0)	Lasso (alpha = 1)	Elastic Net (alpha = 0.7)
0.04309983	0.04150493	0.04204704

Dimension Reduction

PCR

Let's see if by reducing the dimension we can improve the performance.



As we can see on the graph above and according to the principal component analysis, to get the more information, we should keep the all 8 components.

Thus, we get :

PRC	Result
components	8
MSE	0.04150581

Data normalization

In certain cases, normalizing the data can bring improvements on the performance of the model. The tests done above, were also done on normalized data but did not bring significant improvement. That's why we judge not necessary to add more results because these are predictable: normalization being a simple linear transformation, regression learns it as needed.

Models

Linear Regression model LM	KNN model KNN (K=10)
MSE = 0.04150581	MSE = 0.01599222

The KNN model performed well as the mse dropped from 0.04 to 0.01.

As the hypotheses for linearity were not valid for this dataset, moving beyond linearity should bring significant improvement in the performance of the models.

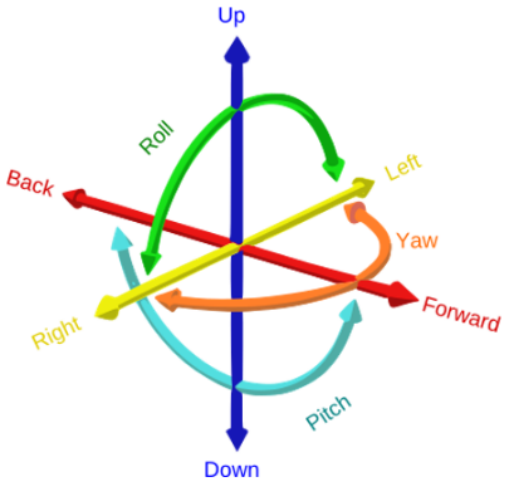
Multinomial Regression

The multinonial regression is quite limited as for the models set up, we will focus more on the splines, in particular the additive models.

Splines

In the case of a multidimensional model, the basic, natural and smooth splines do not apply. We can use the tensors but the dimension increases exponentially and we cruelly lose in interpretability. To overcome this problem, GAMs (Generalized Additive Models) are used. This model is good at explaining but often remains less efficient at predicting.

As the task is on kinematics of robot arm, the degree of freedom here mentions the motion capability of the arm. The degree of freedom is defined as the way in which the robot arm can move.



The arm has 6 D.O.F (Degree of Freedom) in space but due to the formation of linkage one or more D.O.F is lost due to the presence of constraint on the arm.

Thus for GAM, df = 6.

function	Degree of Freedom	MSE
ns()	not specified	0.04047486
ns()	6	0.04010025
s()	6	0.04028447

As said above, GAMs are less effective at predicting. The performance based on the mse values is quite the same as for linear models tested before.

SVR (The chosen model)

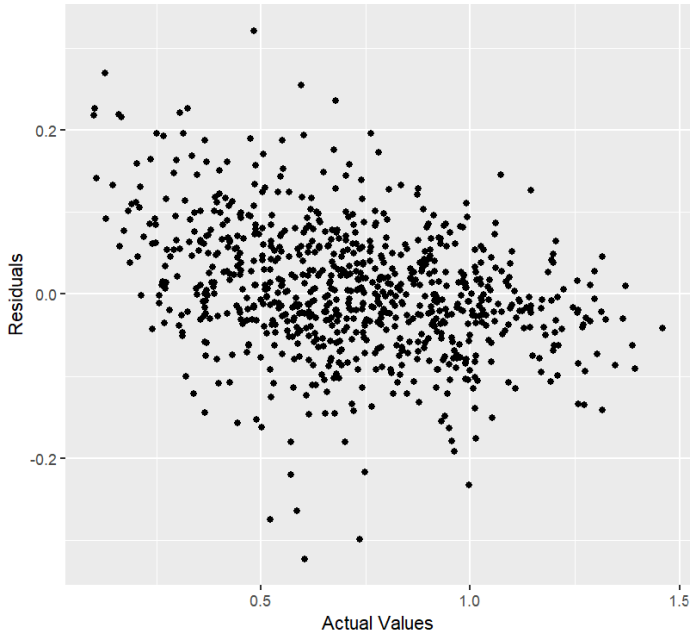
SVR models perform well on a wide range of data, especially when the data has large number of features or when the data is not linearly separable like in our case. Thus they can handle high dimensional data well thanks to the kernel trick and are more interpretable.

We use 'svmRadial' function as the kernel function as it uses a radial basis function kernel.

We define a range of of values for the cost C and sigma and use the expand.grid function to create a grid of all possible combinations of C and sigma values.

After training, the best C and sigma for this model are 10 and 0.1 respectively and the model performed very well with an mse of 0.006160302

We plot the residual plot to assess the model



It shows randomly distributed points around the x-axis indicating that the errors are randomly distributed and not systematically biased.

Conclusion

After a study of different models depending on the structure of our data, it turns out that the SVR-based model gives a minimum MSE of 0.006160302.