Machine Learning - 1100-ML0ENG (Ćwiczenia informatyczne Z-23/24)

Modeling the strength of concrete

The dataset contains data describing the materials used in the production of concrete. Concrete can have different **strength**s (variable target). **Cement, slag, ash, water, superplasticiser, fine aggregate and coarse aggregate, as well as ageing time (measured in days)** were used as components for the production of concrete.

```
# preparation of the dataset ------------------------------------------------
```

```
#concrete <- read.csv("http://imul.math.uni.lodz.pl/~bartkiew/ml/data/concrete.csv")
```

```
concrete <- read.csv("concrete.csv")
```

```
summary(concrete)
```

```
concrete.n <- as.data.frame(lapply(concrete, normalize))
```

```
# training/test -------------------------------------------------------------
```

```
library(caret)
set.seed(123)
inTrain <- createDataPartition(y=concrete.n$strength, p=0.75, list=FALSE)
cn_train <- concrete.n[inTrain,]
cn_test <- concrete.n[-inTrain,]
```

```
# building a network model - neuralnet package ----------------------
------
```

```
# # first model ------------------------------------------------------------
```

```
library(neuralnet)
set.seed(2023)
model <- neuralnet(strength ~
cement+slag+ash+water+superplastic+coarseagg+fineagg+age,
 data = cn_train)
plot(model)
```

In the example above, a network model was built and then the network was drawn. In this simple model, we have one input node for the data in a set of eight variables and a single hidden node and a single output node that predicts the bond strength of the concrete. At the bottom of the image we have the number of steps and the error (SSE - Sum of Squared Errors) plotted. Obviously, lower values allow you to select a better model.

```
res <- compute(model, cn_test)
predicted_strength <- res$net.result
```

```
MSE(predicted_strength, cn_test$strength)
r2(predicted_strength, cn_test$strength)
```

Since we are dealing with a regresion task and not a classification task, so we will use MSE error or R^2 and not a confusion matrix.

```
plot(predicted_strength, cn_test$strength,
 col="blue", main="test set neuralnet", pch=16)
abline(0,1)
```

```
# second model with 8 nodes in the hidden layer --------------------------
```

```
model2 <- neuralnet(strength ~cement+slag+
 ash+water+superplastic+coarseagg+fineagg+age, data = cn_train, hidden = 8)
plot(model2)
wyniki2 <- compute(model2, cn_test)
predicted_strength2 <- wyniki2$net.result
```

```
MSE(predicted_strength2, cn_test$strength)
r2(predicted_strength2, cn_test$strength)
```

```
plot(predicted_strength2, cn_test$strength)
abline(0,1)
```

Neural network nnet

nnet

```
install.packages("nnet")
library(nnet)
```

```
# network model built using the nnet package ------------------------
------
```

```
set.seed(2023)
concrete.nn = nnet(strength ~ ., data = cn_train, size = 1)
concrete.nn
summary(concrete.nn)
```

```
#install.packages("NeuralNetTools")
library(NeuralNetTools)
plotnet(concrete.nn , alpha=0.6)
```

```
cn.predict = predict(concrete.nn, cn_test)
MSE(cn.predict, cn_test$strength)
r2(cn.predict, cn_test$strength)
```

```
plot(cn.predict, cn_test$strength,
 col="tomato", main="test_set nnet", pch=16)
abline(0,1)
```

```
# caret package control ----------------------------------------------------
```

```
control = trainControl(method="cv", number=10)
grid=expand.grid(.decay=c(0.0005,0.1), .size=c(7,8,9))
model3 = train(strength ~ ., data = cn_train,
 method='nnet', tuneGrid=grid,
 trControl=control)
plot(model3)
model3
```

Last modified: czwartek, 25 stycznia 2024, 10:36

Platformą administruje Komisja ds. Doskonalenia Dydaktyki wraz z Centrum Informatyki Uniwersytetu Łódzkiego Więcej

Na platformie jest wykorzystywana metoda logowania za pośrednictwem Centralnego Systemu Logowania.

Studentów i pracowników Uniwersytetu Łódzkiego obowiązuje nazwa użytkownika i hasło wykorzystywane podczas logowania się do systemu USOSweb.