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

neuralnet - Space Shuttle

A set of data from the simulation of the use of the shuttle space from the MASS package.

https://stat.ethz.ch/R-manual/R-devel/library/MASS/html/shuttle.html

```
#install.packages("MASS")
library(MASS)
```

The data is in the MASS package.

```
data(shuttle)
str(shuttle)
```

Notice that all of the variables are categorical and the response is use with two levels, **auto** and **noauto.**

- stability: This is stable positioning or not (stab/xstab)
- error: This is the size of the error (MM / SS / LX)
- sign: This is the sign of the error, positive or negative (pp/nn)
- wind: This is the wind sign (head / tail)
- magn: This is the wind strength (Light / Medium / Strong / Out of Range)
- vis: This is the visibility (yes / no)

Preparing the data for a neural network is very important as all the covariates and responses **need to be numeric**. In our case, all of the input features are categorical. The caret package allows us to quickly create dummy variables as our input features.

```
dummies <- dummyVars(use ~ .,shuttle, fullRank = T)
dummies
```

```
shuttle.2 = as.data.frame(predict(dummies, newdata=shuttle))
names(shuttle.2)
str(shuttle.2)
```

Because use was response in dummy coding, we must add it to data set.

```
shuttle.2$use <- ifelse(shuttle$use == "auto", 1, 0)
table(shuttle.2$use)
```

Now we can create train and tet sets.

The caret package also provides us with the functionality to create the train and test sets. The idea is to index each observation as train or test and then split the data accordingly. Let's do this with a 70/30 train to test split - **this is the third way to split a data set.**

```
set.seed(45)
trainIndex <- createDataPartition(shuttle.2$use, p = .7, list =FALSE)
```

The values in trainIndex provide us with the row number; in our case, 70 per cent of the total row numbers in shuttle.2.

```
shuttleTrain <- shuttle.2[trainIndex, ]
shuttleTest <- shuttle.2[-trainIndex, ]
```

Modeling and evaluation

Because we need formula such as y~x1+x2+x3+x4, we can produce it, using **as.formula()** function

```
n <- names(shuttleTrain)
formula <- as.formula(paste("use ~", paste(n[!n %in% "use"],collapse = " + ")))
formula
```

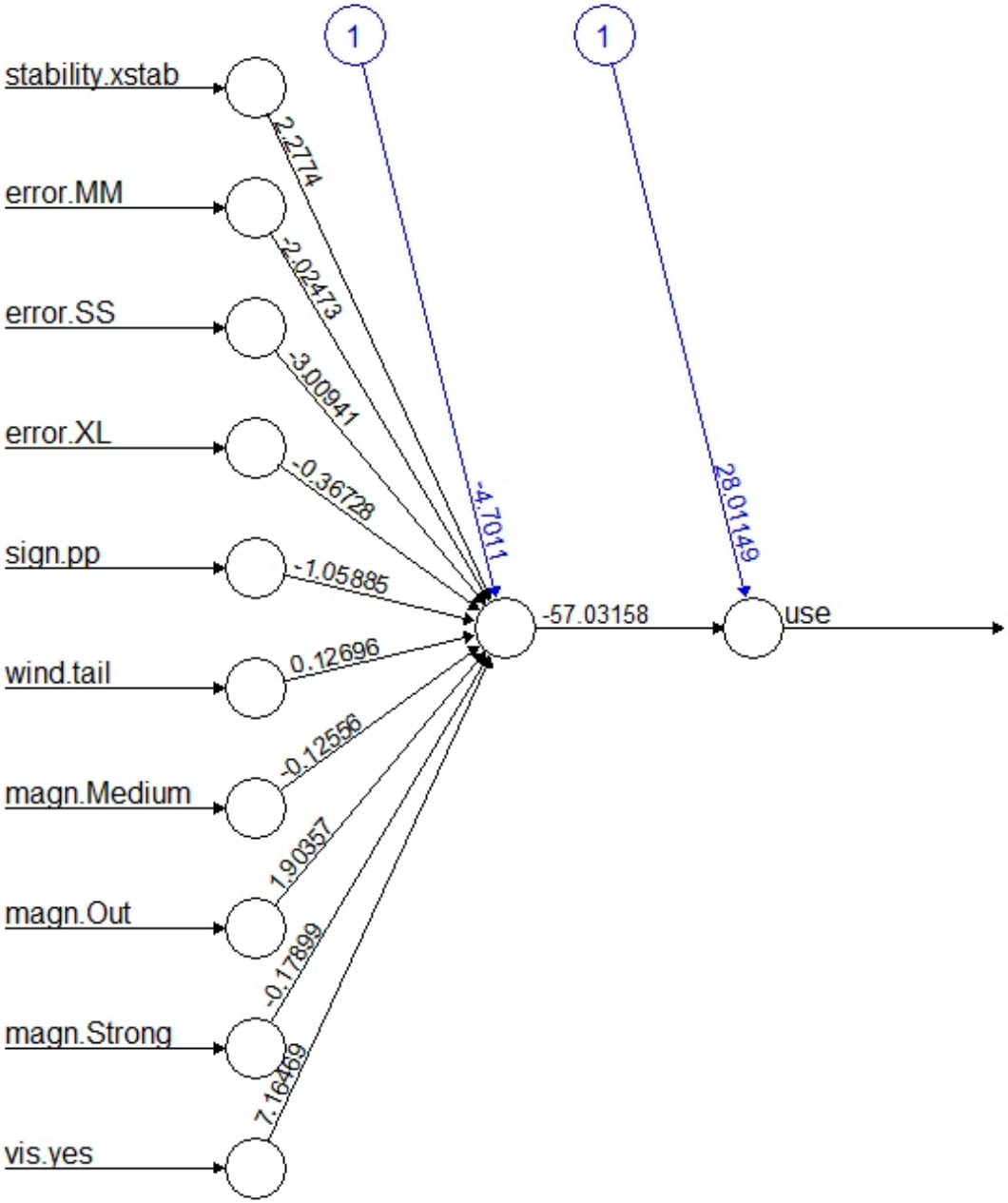
use ~ stability.xstab + error.MM + error.SS + error.XL + sign.pp + wind.tail + magn.Medium + magn.Out + magn.Strong + vis.yes

In **neuralnet function**, we also have parameters

- **act.fct -** this is the activation function with the default logistic and tanh available
- **err.fct -** this is the function used to calculate the error with the default **sse**(sum of squared) as we are dealing with binary outcomes, we use **ce for cross-entropy**
- **linear.output -** this is a logical argument on whether or not to ignore act.fct with the default TRUE, so for our data, this will need to be FALSE

```
net1 <- neuralnet(formula, data = shuttleTrain, err.fct = "ce", linear.output = FALSE)
```

```
net1
plot(net1)
```



Error: 0.011398 Steps: 577

We can noticed, that the error is extremely low at 0.0099. Now, we can check, how it does on the test set:

```
res2 <- compute(net1, shuttleTest[, 1:10])
predTest2 <- res2$net.result
predTest2
```

These results are numeric, so let's turn them into 0 or 1 and follow this up with a confusion matrix:

```
predTest2 <- ifelse(predTest2 >= 0.5, 1, 0)
table(predTest2, shuttleTest$use)
```

```
> table(predTest2, shuttleTest$use)

predTest2  0  1
        0 32  0
        1  1 43
```

Only one false positive in the test set. If you wanted to identify which one this was, use the which() function to single it out.

```
which(predTest2 == 1 & shuttleTest$use == 0)
```

It is 60, so

```
shuttleTest[60,]
shuttle[203,]
```

> shuttle[60,]

```
> shuttleTest[60,]
    stability.xstab error.MM error.SS error.XL sign.pp wind.tail magn.Medium
203               0        1        0        0       1         0           0
> shuttle[203,]
    stability error sign wind   magn vis    use
203      stab    MM   pp head Strong yes noauto
>
```

So our network says use auto control and the astronaut has used manual control.

Last modified: piątek, 20 stycznia 2023, 11:35

Accessibility settings

Przetwarzanie danych osobowych

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

Informacje na temat logowania

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.

Deklaracja dostępności