Mathematics for Big Data - Exercise 3

Lawrence Adu-Gyamfi (1484610)

06/06/2019

Contents

1 INTRODUCTION **2**
1.1 Exploring the rock dataset 2

2 QUESTION 1 **3**
2.1 Question 3
2.2 Solution 3

3 QUESTION 2 **5**
3.1 Question 5
3.2 Solution 5

4 QUESTION 3 **7**
4.1 Question 7
4.2 Solution 7

5 QUESTION 4 **8**
5.1 Question 8
5.2 Solution 8

6 Appendix **10**
6.1 Source Code 10

1 INTRODUCTION

This report presents the code and the results of analysis and predicting the permeability using the variables in the rock dataset from the MASS package.

The object rock from the library MASS contains different measurements on 48 rock samples from a petroleum reservoir. The response variable is the permeability of those rocks and the explanatory variables are the remaining variables.

1.1 Exploring the rock dataset

The following shows some details about the rock dataset.

Here we confirm the dataset indeed does have 48 observations for 4 predictors.

```
## [1] 48  4
```

Below is a sample of the dataset showing the first 6 observations.

area	peri	shape	perm
4990	2791.90	0.0903296	6.3
7002	3892.60	0.1486220	6.3
7558	3930.66	0.1833120	6.3
7352	3869.32	0.1170630	6.3
7943	3948.54	0.1224170	17.1
7979	4010.15	0.1670450	17.1

And we we see a summary of some statistics of the dataset grouped by predictor.

```
##       area            peri           shape             perm
##  Min.   : 1016   Min.   : 308.6   Min.   :0.09033   Min.   :   6.30
##  1st Qu.: 5305   1st Qu.:1414.9   1st Qu.:0.16226   1st Qu.:  76.45
##  Median : 7487   Median :2536.2   Median :0.19886   Median : 130.50
##  Mean   : 7188   Mean   :2682.2   Mean   :0.21811   Mean   : 415.45
##  3rd Qu.: 8870   3rd Qu.:3989.5   3rd Qu.:0.26267   3rd Qu.: 777.50
##  Max.   :12212   Max.   :4864.2   Max.   :0.46413   Max.   :1300.00
```

2 QUESTION 1

2.1 Question

Fit a 3-3-1 neural networks to model the permeability of the rocks (perm) taking as inputs the variables (area, peri and shape). Evaluate the goodness of the fit using the determination coefficients.

2.2 Solution

Below is the summary of fitting a 3-3-1 (3 neurons in the input layer, 3 in the hidden layer and 1 for the output) neural network model on the dataset.

```
## # weights:  16
## initial  value 17298262.100406
## final  value 9009185.560000
## converged
```

```
## a 3-3-1 network with 16 weights
## options were - linear output units
##  b->h1 i1->h1 i2->h1 i3->h1
##   0.01  -0.37  -0.31   0.42
##  b->h2 i1->h2 i2->h2 i3->h2
##  -0.09  -0.02   0.42   0.15
##  b->h3 i1->h3 i2->h3 i3->h3
##  -0.36   0.45  -0.31  -0.65
##   b->o  h1->o  h2->o  h3->o
## 138.64   0.14 137.90 138.91
```

Below is the RSS (Residuals Squared Sum) for the fitting the neural network on the dataset as it is:

```
rock.nnet$value
```

```
## [1] 9009186
```

which is not very different from the TSS of the null model:

```
(rock_TSS <- sum((perm - mean(perm))^2))
```

```
## [1] 9009186
```

The performance of the model is shown below which is denoted the amont of variance in the dataset captured by the model

```
(R2.nnet <- 1 - rock.nnet$value / rock_TSS)
```

```
## [1] 0
```

We confirm indeed for now the model does not perform ver well with an R-squared value of approximately 0.

2.2.1 Fitting a Linear Model

For the sake of comparison we fit a linear model to estimate the performance of the neural network model.

```
rock.lm <- lm(perm ~ shape+area+peri, rock)
(RSS.lm <- sum((perm - predict(rock.lm))^2))
```

```
## [1] 2663023
```

```
(R2.lm <- 1 - (RSS.lm / rock_TSS))
```

```
## [1] 0.7044103
```

Comparing the RSS and the determination coefficients of both models we realise the linear model performs much better on the data than the typical neural network model constructed in this way.

3 QUESTION 2

3.1 Question

Ripley (1997) proposed fitting a neural network to this data with some previous transformations: log-scaling the permeability and dividing the predictors area and peri by 10000 units. Fit again the neural network and compare the results.

3.2 Solution

As demanded by the question, the permeability values are log-scaled, while the area and perimeter values are divided by 10000 as shown below.

3.2.1 Scaling of rock dataset

```
scaled_rock = rock
scaled_rock$perm <- log(rock$perm)
scaled_rock$area <- rock$area / 10000
scaled_rock$peri <- rock$peri / 10000

head(scaled_rock)
```

area	peri	shape	perm
0.4990	0.279190	0.0903296	1.840550
0.7002	0.389260	0.1486220	1.840550
0.7558	0.393066	0.1833120	1.840550
0.7352	0.386932	0.1170630	1.840550
0.7943	0.394854	0.1224170	2.839078
0.7979	0.401015	0.1670450	2.839078

A new model is fitted to the scaled dataset, and the results of this model are shown below:

3.2.2 Fitting Neural Network Model to Scaled Dataset

```
## # weights:  16
## initial  value 1409.980332
## iter  10 value 71.330325
## iter  20 value 33.581208
## iter  30 value 24.920631
## iter  40 value 24.095828
```

```
## iter  50 value 23.995093
## iter  60 value 23.231532
## iter  70 value 22.198966
## iter  80 value 21.089566
## iter  90 value 21.072198
## iter 100 value 21.070715
## final  value 21.070715
## stopped after 100 iterations
```

```
## [1] 21.07071
```

Using this approach a very significant determination coefficient value is obtained as shown above.

```
(R2_scaled_rock <- 1 - scaled_rock.nnet$value / scaled_rock.TSS)
```

```
## [1] 0.8340002
```

4 QUESTION 3

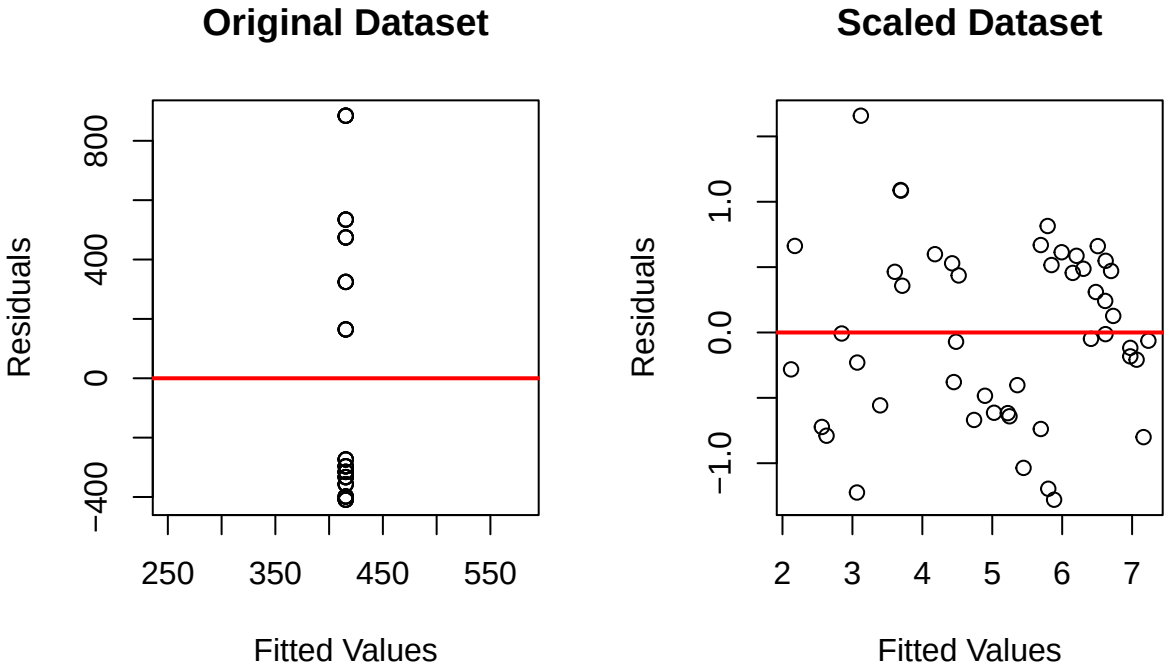
4.1 Question

Obtain a graphical representation to visualize the residuals of the model and the fitted values.

4.2 Solution

In this section, the residuals and fitted values of each of the models are compared.

The figure below compares the residuals of the two models using the unscaled and scaled data respectively.



The above graphs show much better behaved residuals when the data is scaled as recommended by Ripley.

5 QUESTION 4

5.1 Question

Assess the stability of the estimations by running the numerical algorithm 100 times.

5.2 Solution

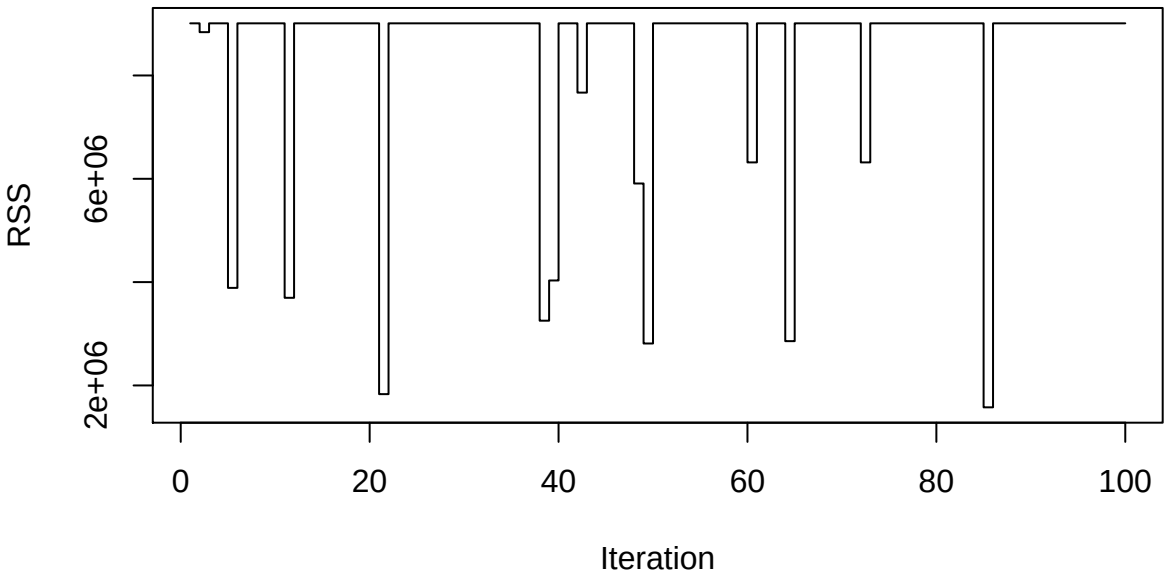
In this section, the two models are run for 100 times and their subsequent results are compared.

Below are the results:

Below is the plot of the RSS for the original dataset (without scaling) run over a 100 iterations.

The determination coefficient is shown below:

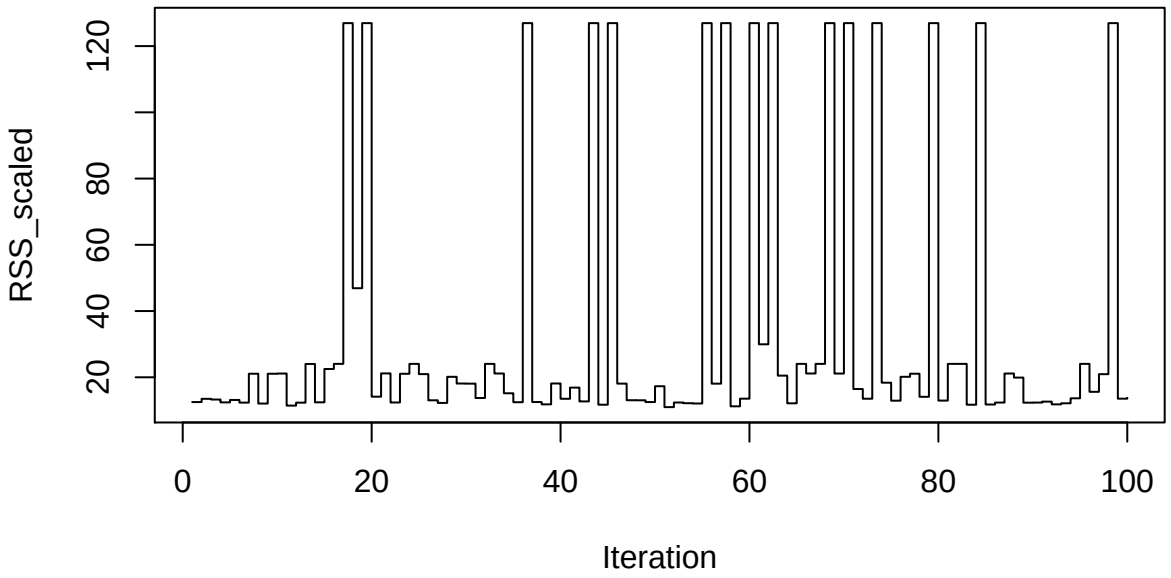
```
## [1] 0.8250718
```



The determination coefficient is shown below:

```
## [1] 0.9134112
```

Below is the plot of the RSS for the dataset with scaling, run over a 100 iterations.



Comparing the R_squared values for both models, it is evident that the model based on the scaled dataset performs much better, even if the value for the model on the original dataset is significant and improved.

From the plots of the residuals over the different iterations,we notice the original model is able to obtain the lowest RSS on few occassions compared to the scaled model which rather for most of the iterations has a lower RSS value.

6 Appendix

6.1 Source Code

```
library(MASS)
library(nnet)
attach(rock)

dim(rock)
summary(rock)
head(rock)


rock.nnet <- nnet(perm ~ area + peri + shape, size=3, linout=T)
summary(rock.nnet)
rock.nnet$residuals
rock.nnet$value
rock.nnet$fitted.values

rock_TSS <- sum((perm - mean(perm))^2)
R2.nnet <- 1 - rock.nnet / rock_TSS

rock.lm <- lm(perm ~ shape+area+peri, rock)
RSS.lm <- sum((perm - predict(rock.lm))^2)
R2.lm <- 1 - (RSS.lm / rock_TSS)


scaled_rock = rock
scaled_rock$perm <- log(rock$perm)
scaled_rock$area <- rock$area / 10000
scaled_rock$peri <- rock$peri / 10000

head(scaled_rock)


scaled_rock.nnet <- nnet(perm ~ area + shape + peri, scaled_rock, size=3, linout=T)
scaled_rock.nnet$value
scaled_rock.TSS <- sum((scaled_rock$perm - mean(scaled_rock$perm))^2)
R2_scaled_rock <- 1 - scaled_rock.nnet$value / scaled_rock.TSS


RSS <- NULL
RSS_scaled <- NULL
```

```
best.RSS <- rock.nnet$value
best.RSS_scaled <- scaled_rock.nnet$value

for (i in 1:100){
  aux.nnet <- nnet(perm ~ area+shape+peri, rock, size=3, linout=T)
  RSS[i] <- aux.nnet$value
  if (aux.nnet$value < best.RSS)
    {
    rock.nnet <- aux.nnet
    best.RSS <- rock.nnet$value
  }
  }

for (i in 1:100){
  aux.nnet <- nnet(perm ~ area+shape+peri, scaled_rock, size=3, linout=T)
  RSS_scaled[i] <- aux.nnet$value
  if (aux.nnet$value < best.RSS_scaled){
    scaled_rock.nnet <- aux.nnet
    best.RSS_scaled <- scaled_rock.nnet$value}
  }

R2.nnet <- 1 - rock.nnet$value / TSS
R2_scaled_rock <- 1 - scaled_rock.nnet$value / scaled_rock.TSS
plot(1:100, RSS, type="s", xlab="Iteration")
plot(1:100, RSS_scaled, type="s", xlab="Iteration")
```