**Praktikum 2**

**Data Pre-Processing**

**Tahapan Data Pre-processing steps**

1. Penanganan missing value
2. Split data menjadi data training dan data uji
3. Scaling

**Dataset**

## Country Age Salary Purchased

## 1 France 44 72000 No

## 2 Spain 27 48000 Yes

## 3 Germany 30 54000 No

## 4 Spain 38 61000 No

## 5 Germany 40 **NA**  Yes

## 6 France 35 58000 Yes

## 7 Spain **NA**  52000 No

## 8 France 48 79000 Yes

## 9 Germany 50 83000 No

## 10 France 37 67000 Yes

1. **Penanganan missing value**

# By replacing it to the average value for non NA entries.

dataset$Age <- ifelse(is.na(dataset$Age),

ave(dataset$Age, FUN = function(x)

mean(x, na.rm = TRUE)),

dataset$Age)

dataset$Salary <- ifelse(is.na(dataset$Salary),

ave(dataset$Salary, FUN = function(x)

mean(x, na.rm = TRUE)),

dataset$Salary)

dataset

## Country Age Salary Purchased

## 1 France 44.00000 72000.00 No

## 2 Spain 27.00000 48000.00 Yes

## 3 Germany 30.00000 54000.00 No

## 4 Spain 38.00000 61000.00 No

## 5 Germany 40.00000 63777.78 Yes

## 6 France 35.00000 58000.00 Yes

## 7 Spain 38.77778 52000.00 No

## 8 France 48.00000 79000.00 Yes

## 9 Germany 50.00000 83000.00 No

## 10 France 37.00000 67000.00 Yes

1. Splitting into training and test dataset : When the dataset is presented to us to do machine learning stuff we need some data as part of training and some to test the model after the learning stage is done.

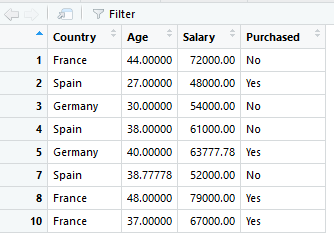
#install.packages("caTools") #if not present

library(caTools) #adding caTools to the library

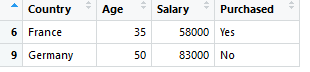
set.seed(123) # this is to ensure same output as split is done randomly, you can exclude in real time

split = sample.split(dataset$Purchased,SplitRatio = 0.8)

training\_set = subset(dataset,split == TRUE)



test\_set = subset(dataset, split == FALSE)



SplitRatio is the ratio in which training and test set, its usually set an 80:20 for training and test respectively.

sample.split() methid takes the column and calculates a numeric array with true and false in random locations and with the given split ratio.

subset() method takes the dataset and subset according to the condition

1. **Feature Scaling :**

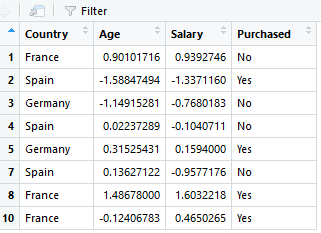
Feature scaling is needed when different features has different ranges, for example Age and Salary Column.

#feature scaling

training\_set[,2:3] = scale(training\_set[,2:3])

test\_set[,2:3] = scale(test\_set[,2:3])

2:3 is for both Age and Salary Now the dataset(training and test both) looks like :

training\_set

## Country Age Salary Purchased

## 1 1 0.90101716 0.9392746 0

## 2 2 -1.58847494 -1.3371160 1

## 3 3 -1.14915281 -0.7680183 0

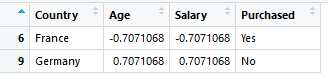
## 4 2 0.02237289 -0.1040711 0

## 5 3 0.31525431 0.1594000 1

## 7 2 0.13627122 -0.9577176 0

## 8 1 1.48678000 1.6032218 1

## 10 1 -0.12406783 0.4650265 1

test\_set

## Country Age Salary Purchased

## 6 1 -0.7071068 -0.7071068 1

## 9 3 0.7071068 0.7071068 0

**Data Pre-Processing With Caret in R**

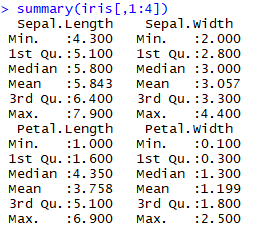
**1. Scale**

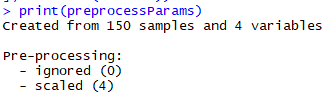
The scale transform calculates the standard deviation for an attribute and divides each value by that standard deviation.

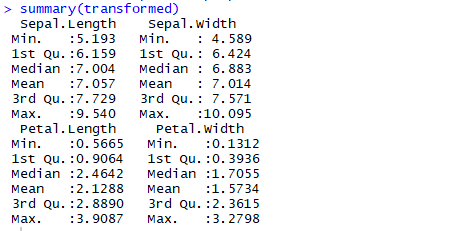
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | # load libraries  library(caret)  # load the dataset  data(iris)  # summarize data  summary(iris[,1:4])  # calculate the pre-process parameters from the dataset  preprocessParams <- preProcess(iris[,1:4], method=c("scale"))  # summarize transform parameters  print(preprocessParams)  # transform the dataset using the parameters  transformed <- predict(preprocessParams, iris[,1:4])  # summarize the transformed dataset  summary(transformed) |

Output :

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | Sepal.Length    Sepal.Width     Petal.Length    Petal.Width  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  Median :5.800   Median :3.000   Median :4.350   Median :1.300  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500    Created from 150 samples and 4 variables    Pre-processing:    - ignored (0)    - scaled (4)      Sepal.Length    Sepal.Width      Petal.Length     Petal.Width  Min.   :5.193   Min.   : 4.589   Min.   :0.5665   Min.   :0.1312  1st Qu.:6.159   1st Qu.: 6.424   1st Qu.:0.9064   1st Qu.:0.3936  Median :7.004   Median : 6.883   Median :2.4642   Median :1.7055  Mean   :7.057   Mean   : 7.014   Mean   :2.1288   Mean   :1.5734  3rd Qu.:7.729   3rd Qu.: 7.571   3rd Qu.:2.8890   3rd Qu.:2.3615  Max.   :9.540   Max.   :10.095   Max.   :3.9087   Max.   :3.2798 |

**Output**





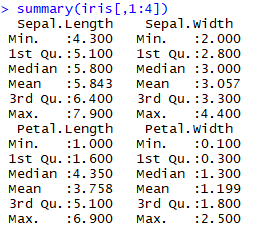
**2. Center**

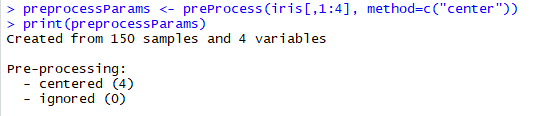
The center transform calculates the mean for an attribute and subtracts it from each value.

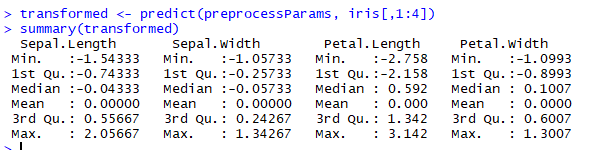
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | # load libraries  library(caret)  # load the dataset  data(iris)  # summarize data  summary(iris[,1:4])  # calculate the pre-process parameters from the dataset  preprocessParams <- preProcess(iris[,1:4], method=c("center"))  # summarize transform parameters  print(preprocessParams)  # transform the dataset using the parameters  transformed <- predict(preprocessParams, iris[,1:4])  # summarize the transformed dataset  summary(transformed) |

Output :

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | Sepal.Length    Sepal.Width     Petal.Length    Petal.Width  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  Median :5.800   Median :3.000   Median :4.350   Median :1.300  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500    Created from 150 samples and 4 variables    Pre-processing:    - centered (4)    - ignored (0)    Sepal.Length       Sepal.Width        Petal.Length     Petal.Width  Min.   :-1.54333   Min.   :-1.05733   Min.   :-2.758   Min.   :-1.0993  1st Qu.:-0.74333   1st Qu.:-0.25733   1st Qu.:-2.158   1st Qu.:-0.8993  Median :-0.04333   Median :-0.05733   Median : 0.592   Median : 0.1007  Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.000   Mean   : 0.0000  3rd Qu.: 0.55667   3rd Qu.: 0.24267   3rd Qu.: 1.342   3rd Qu.: 0.6007  Max.   : 2.05667   Max.   : 1.34267   Max.   : 3.142   Max.   : 1.3007 |







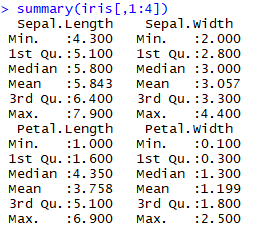
**3. Standardize**

Combining the scale and center transforms will standardize your data. Attributes will have a mean value of 0 and a standard deviation of 1.

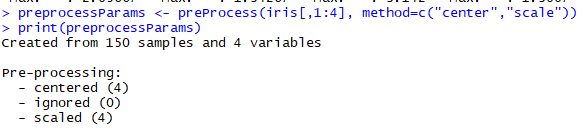
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | # load libraries  library(caret)  # load the dataset  data(iris)  # summarize data  summary(iris[,1:4])  # calculate the pre-process parameters from the dataset  preprocessParams <- preProcess(iris[,1:4], method=c("center", "scale"))  # summarize transform parameters  print(preprocessParams)  # transform the dataset using the parameters  transformed <- predict(preprocessParams, iris[,1:4])  # summarize the transformed dataset  summary(transformed) |

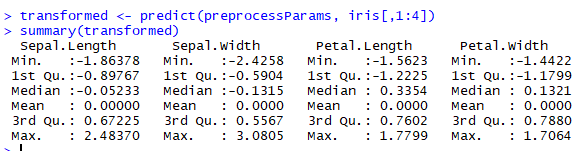
Output :

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | Sepal.Length    Sepal.Width     Petal.Length    Petal.Width  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  Median :5.800   Median :3.000   Median :4.350   Median :1.300  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500    Created from 150 samples and 4 variables    Pre-processing:    - centered (4)    - ignored (0)    - scaled (4)    Sepal.Length       Sepal.Width       Petal.Length      Petal.Width  Min.   :-1.86378   Min.   :-2.4258   Min.   :-1.5623   Min.   :-1.4422  1st Qu.:-0.89767   1st Qu.:-0.5904   1st Qu.:-1.2225   1st Qu.:-1.1799  Median :-0.05233   Median :-0.1315   Median : 0.3354   Median : 0.1321  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000  3rd Qu.: 0.67225   3rd Qu.: 0.5567   3rd Qu.: 0.7602   3rd Qu.: 0.7880  Max.   : 2.48370   Max.   : 3.0805   Max.   : 1.7799   Max.   : 1.7064 |



**4. Normalize**



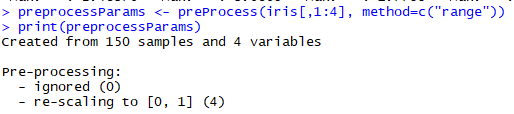


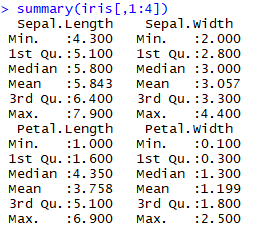
Data values can be scaled into the range of [0, 1] which is called normalization.

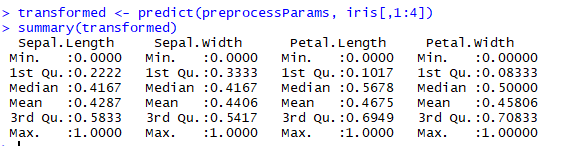
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | # load libraries  library(caret)  # load the dataset  data(iris)  # summarize data  summary(iris[,1:4])  # calculate the pre-process parameters from the dataset  preprocessParams <- preProcess(iris[,1:4], method=c("range"))  # summarize transform parameters  print(preprocessParams)  # transform the dataset using the parameters  transformed <- predict(preprocessParams, iris[,1:4])  # summarize the transformed dataset  summary(transformed) |

Output :

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | Sepal.Length    Sepal.Width     Petal.Length    Petal.Width  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  Median :5.800   Median :3.000   Median :4.350   Median :1.300  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500    Created from 150 samples and 4 variables    Pre-processing:    - ignored (0)    - re-scaling to [0, 1] (4)        Sepal.Length     Sepal.Width      Petal.Length     Petal.Width  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000  1st Qu.:0.2222   1st Qu.:0.3333   1st Qu.:0.1017   1st Qu.:0.08333  Median :0.4167   Median :0.4167   Median :0.5678   Median :0.50000  Mean   :0.4287   Mean   :0.4406   Mean   :0.4675   Mean   :0.45806  3rd Qu.:0.5833   3rd Qu.:0.5417   3rd Qu.:0.6949   3rd Qu.:0.70833  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.00000 |







**Sumber :**

<https://rpubs.com/sidTyson92/329310>

<https://machinelearningmastery.com/pre-process-your-dataset-in-r/>

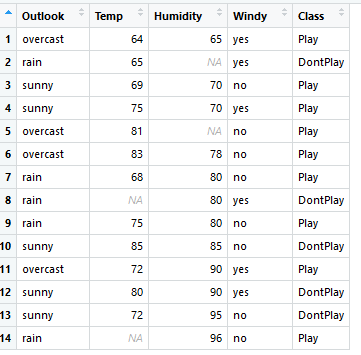
**Tugas**

1. Buat dataset ( original dan yang mengandung missing value)

2. Selipkan 2 missing value pada variable temperature dan humidity secara acak dengan perintah

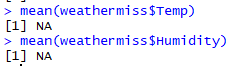
> weathermiss[sample(1:nrow(weathermiss),2),"Temp"]<-NA

> weathermiss[sample(1:nrow(weathermiss),2),"Humidity"]<-NA

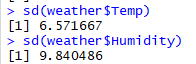
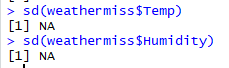


3. Bandingkan rata-rata dan standar deviasi dari data set original dengan data set yang mengandung missing value.

Data original Data missing

Mean



Standar Deviasi

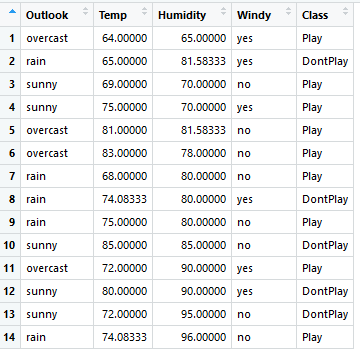
4. ganti missing value dengan nilai mean. Ulangi langkah-langkah tersebut dengan mengganti missing value dengan median dan dilanjutkan menggunakan nilai tertentu.

Data missing diganti mean

> weathermiss$Humidity <- ifelse(is.na(weathermiss$Humidity), ave(weathermiss$Humidity, FUN = function(x) mean(x, na.rm = TRUE)), weathermiss$Humidity)

>

> weathermiss$Temp <- ifelse(is.na(weathermiss$Temp), ave(weathermiss$Temp, FUN = function(x) mean(x, na.rm = TRUE)), weathermiss$Temp)

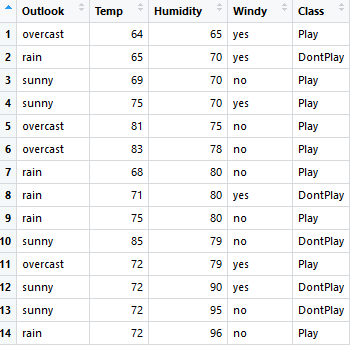


Data missing diganti median

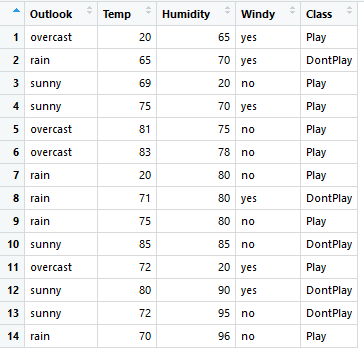
> weathermiss2$Humidity <- ifelse(is.na(weathermiss2$Humidity), ave(weathermiss2$Humidity, FUN = function(x) median(x, na.rm = TRUE)), weathermiss2$Humidity)

>

> weathermiss2$Temp <- ifelse(is.na(weathermiss2$Temp), ave(weathermiss2$Temp, FUN = function(x) median(x, na.rm = TRUE)), weathermiss2$Temp)



Data missing diganti dengan nilai tertentu (20)

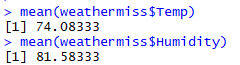
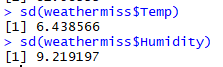


> weathermiss3$Humidity <- ifelse(is.na(weathermiss3$Humidity), 20, weathermiss3$Humidity)

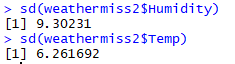
> weathermiss3$Temp <- ifelse(is.na(weathermiss3$Temp), 20, weathermiss3$Temp)

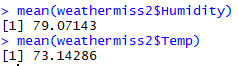
5. Bandingkan rata-rata dan standar deviasinya dengan data original

**Missing value diganti mean**

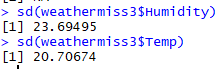
Mean Standar deviasi

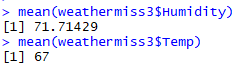
**Missing value diganti median**

Mean Standar Deviasi



**Missing value diganti nilai tertentu (20)**

Mean Standar Deviasi



6. Ulangi langkah ke4 dengan menggunakan library Hmisc

library(Hmisc)





