Final Assignment Models Predictors

Eduard Romero / Carlos Bejarano / Juan Cepeda

2023-05-27

1. Problem definition.

Mobile robotics is a popular solution for exploration of hostile environments (such as toxic or radioactive environments) where a direct human intervention is not possible. In this project it is asked that each team implements a robotic explorer and simulates 3 different environments.

- 1.1 A total of 3 different environments needs to be simulated. Each environment needs to provide at least 3 conditions that can be sensed by the robot.
- 1.1.2 The robot needs to be able of moving from one environment to another.
- 1.1.3 Configuration of the environments (order) must be interchangeable.
- 1.1.4 The robot needs to acquire 3 or more sensor signals that can be use as predictors for supervised algorithms.

1.1 Controlled environments.

- Cold Room.
- Hot Room.
- Toxic Room.

1.1.2 Robot characteristics.

The main features of our robot are as follows:

- MCU Arduino.
- Micromotors DC.
- PCB.
- Battery LIPO 7.4V 300mAh.
- Driver motor TB6612FNG.
- Bluetooth HC06.

1.1.3 Configuration enviroments.

The configuration of our 3 environments will be placed in cascade form one after the other, where our robot will take 50 samples for each environment, each of them will be made of cardboard boxes and conditioned for each situation mentioned.

1.1.4 Configuration Sensors.

Our robot has 3 analog sensors:

- Sensor for air quality measurement MQ135.
- Sensor Humidity DHT11.
- Sensor Temperature LM35(DHT11).

The prediction methods will be as follows :

- KNN.
- Logistic regression.
- Decision tree.
- Lasso regression.

2. Arduino code program.

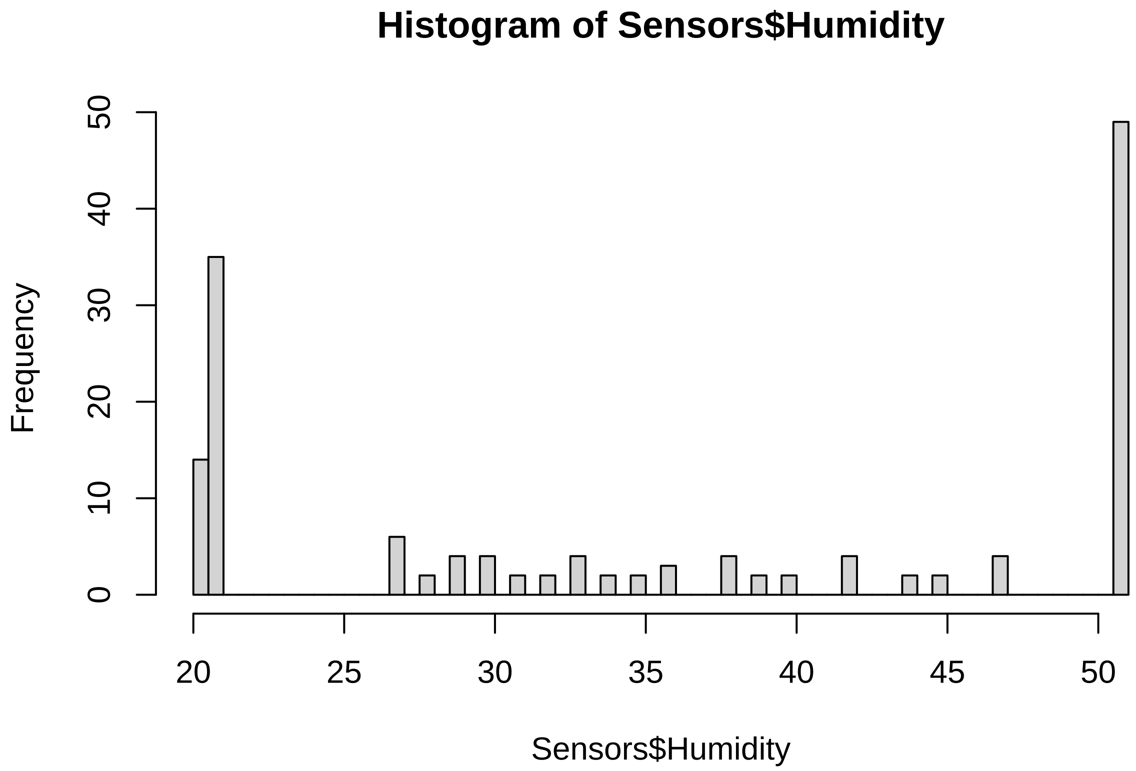
Code Source on repository GitHub

3. Methods for prediction.

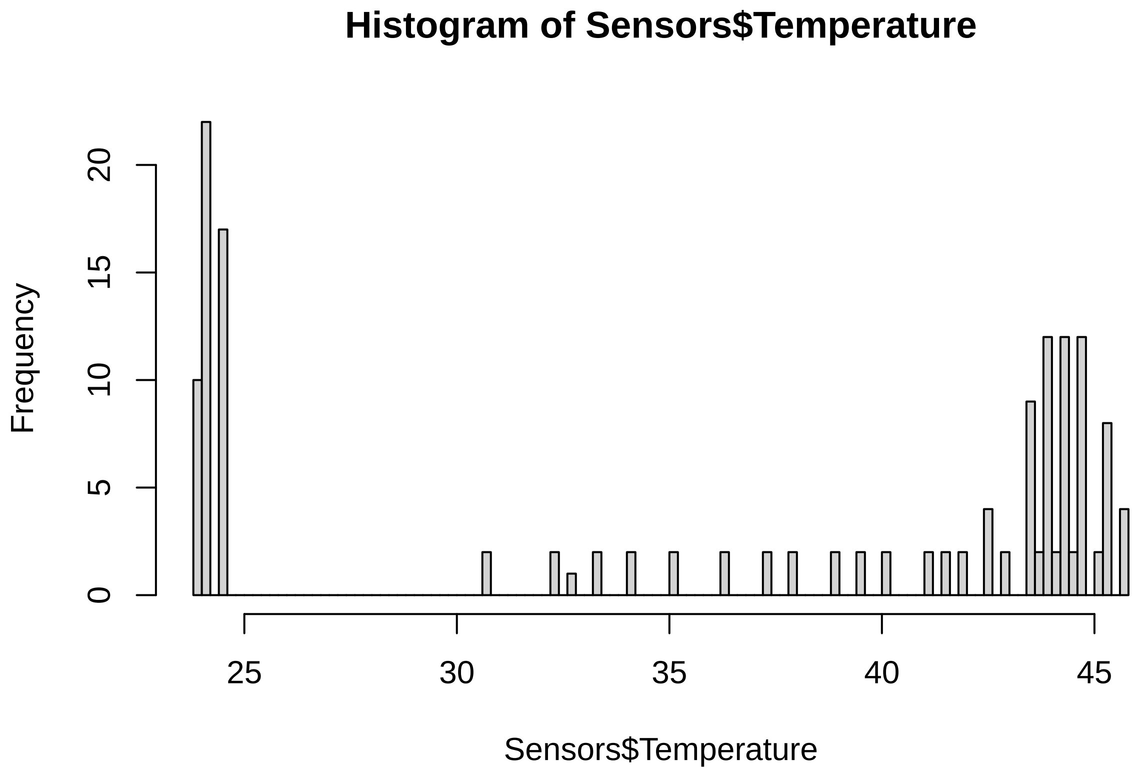
```
# Import the librarys
library (tidyverse)
library (caret)
library(psych)
library(ggplot2)

# Obtain the current folder path and its parent folder path
folder <- dirname(rstudioapi::getSourceEditorContext()$path)
parentFolder <- dirname(folder)

#Read CSV File for trainning
Sensors <- read_csv(file = paste0(parentFolder, "/Datasets/Train_data.csv")) %>% as.data.frame()
#Read CSV File for Predict
DataTest <- read_csv(file = paste0(parentFolder, "/Datasets/Model.csv")) %>% as.data.frame()

# Give our a summary for variables Humidity,Temperatura,PPM,Room
summary(Sensors)
```

```
##     Humidity      Temperature         PPM              Room          
##  Min.   :20.00   Min.   :23.80   Min.   :  9.18   Length:149        
##  1st Qu.:21.00   1st Qu.:24.50   1st Qu.: 17.64   Class :character  
##  Median :34.00   Median :41.60   Median : 19.95   Mode  :character  
##  Mean   :35.68   Mean   :36.28   Mean   :160.63                     
##  3rd Qu.:51.00   3rd Qu.:44.40   3rd Qu.:337.11                     
##  Max.   :51.00   Max.   :45.70   Max.   :575.00                     
```

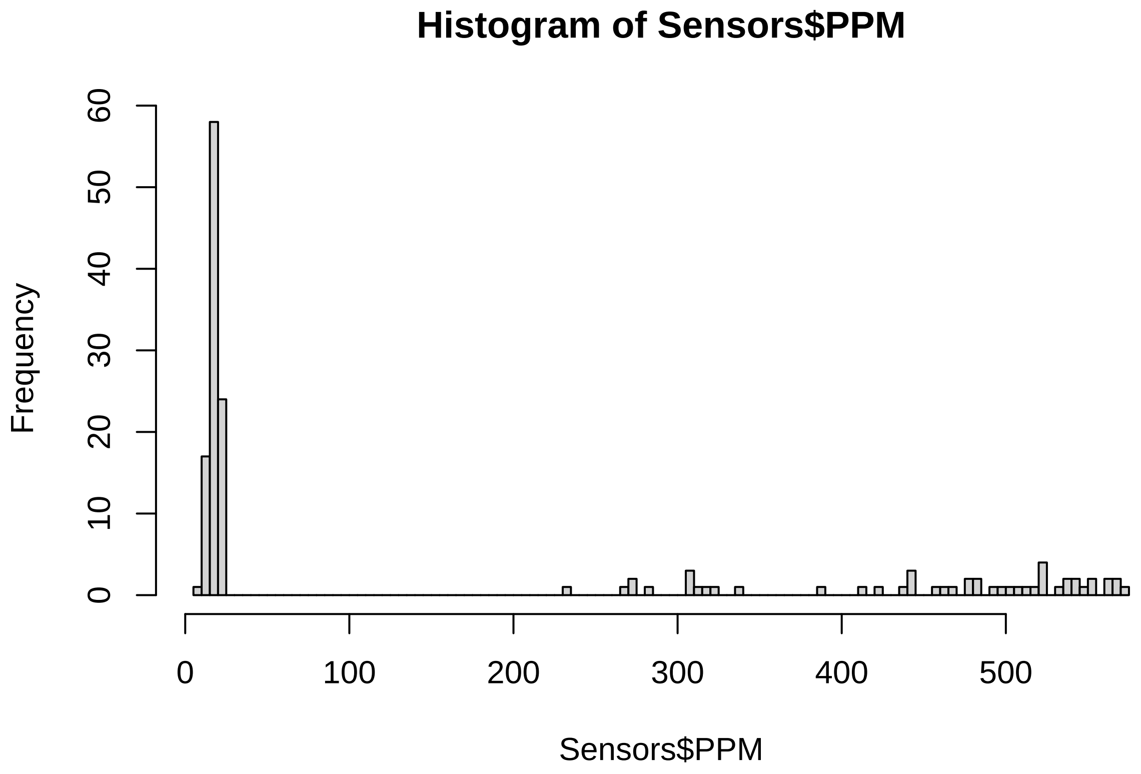
```
# Histogram of the linear model Humidity
hist(Sensors$Humidity,breaks = 100)
```



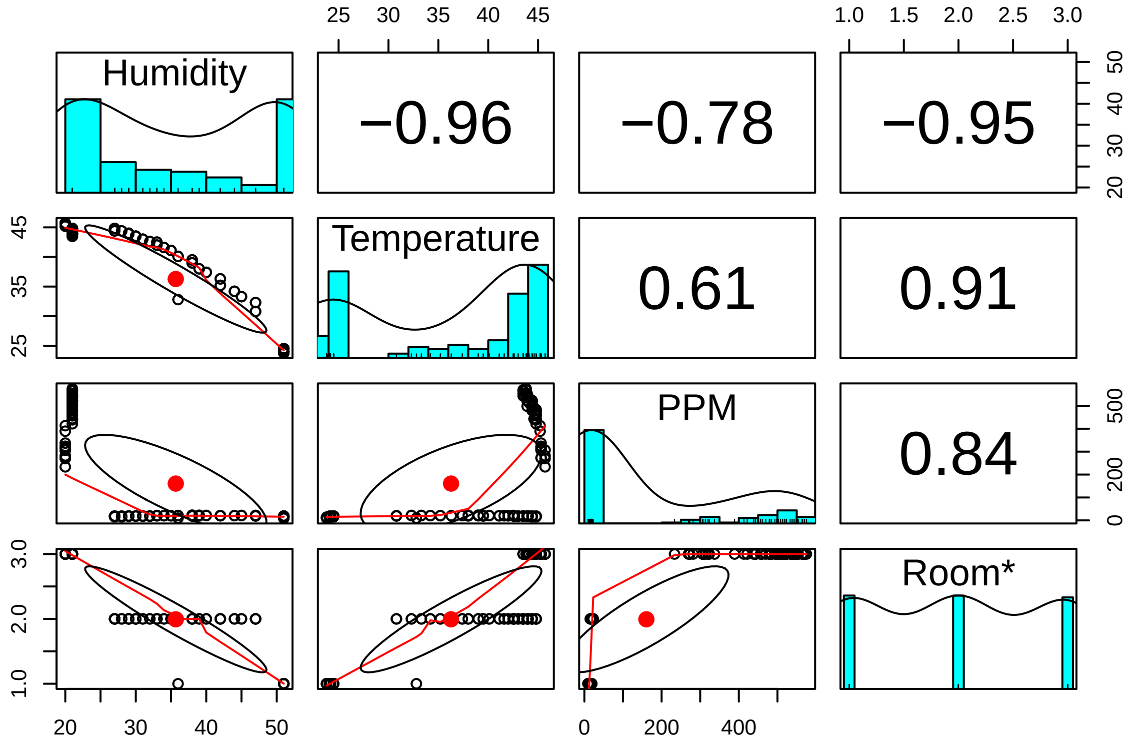
```
# Histogram of the linear model Temperature
hist(Sensors$Temperature,breaks = 100)
```



```
# Histogram of the linear model PPM
hist(Sensors$PPM,breaks = 100)
```



```
pairs.panels(Sensors[c("Humidity",

                       "Temperature",

                       "PPM",

                       "Room")]

             ,pch=21, bg=c("red","green3","blue", "orange")[unclass(Sensors$Room)])
```



```
predictors <- colnames(Sensors)[-3]

sample.index <- sample(1:nrow(Sensors)
                       ,nrow(Sensors)*0.3
                       ,replace = F)

train.data <- Sensors[sample.index,c(predictors,"Room"),drop=F]
test.data <- Sensors[-sample.index,c(predictors,"Room"),drop=F]

# Use 10-Fold cross-validation for all methods

Model <-trainControl(method="cv",number=10)

# Train Model Knn without processing
Model1 <- train(Room~.,data = Sensors,method="knn",trControl=Model)
Model1
```

```
## k-Nearest Neighbors
##
## 149 samples
##   3 predictor
##   3 classes: 'Cold', 'Hot', 'Toxic'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 134, 134, 135, 134, 134, 134, ...
## Resampling results across tuning parameters:
```

```
##
##   k  Accuracy   Kappa
##   5  0.9933333  0.99
##   7  0.9933333  0.99
##   9  0.9933333  0.99
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 9.
```

```
# Predict Model Knn without processing

Predict1 <- predict(Model1,newdata=DataTest,)
Predict1
```

```
##  [1] Cold  Cold  Cold  Cold  Cold  Cold  Cold  Cold  Cold  Cold  Cold  Cold 
## [13] Cold  Cold  Cold  Cold  Cold  Cold  Cold  Cold  Cold  Cold  Cold  Cold 
## [25] Cold  Cold  Cold  Cold  Cold  Cold  Cold  Cold  Cold  Cold  Cold  Hot  
## [37] Hot   Hot   Hot   Hot   Cold  Hot   Cold  Hot   Hot   Hot   Hot   Hot  
## [49] Hot   Hot   Hot   Hot   Hot   Hot   Hot   Hot   Hot   Hot   Hot   Hot  
## [61] Toxic Toxic Toxic Toxic Toxic Toxic Toxic Toxic Toxic Toxic Toxic Toxic
## [73] Toxic Toxic Toxic Toxic Toxic Toxic Toxic Toxic Toxic Toxic Toxic Toxic
## [85] Toxic Toxic Toxic Toxic Toxic Toxic Toxic Toxic Toxic Toxic
## Levels: Cold Hot Toxic
```

```
# Train Model Knn with processing

Model2 <- train(Room~.,data = Sensors,method="knn",preProcess=c("center","scale"),trControl=Model)
Model2
```

```
## k-Nearest Neighbors
##
## 149 samples
##   3 predictor
##   3 classes: 'Cold', 'Hot', 'Toxic'
##
## Pre-processing: centered (3), scaled (3)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 134, 134, 134, 134, 134, 135, ...
## Resampling results across tuning parameters:
##
##   k  Accuracy   Kappa
##   5  0.9933333  0.99
##   7  0.9933333  0.99
##   9  0.9933333  0.99
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 9.
```

```
# Predict Model Knn with processing

Predict2 <- predict(Model2,newdata =DataTest,)
Predict2
```

```
##  [1] Cold Cold Cold Cold Cold Cold Cold Cold Cold Cold Cold Cold Cold Cold Cold
## [16] Cold Cold Cold Cold Cold Cold Cold Cold Cold Cold Cold Cold Cold Cold Cold
## [31] Cold Cold Cold Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot 
## [46] Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot 
## [61] Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot
## [76] Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot  Hot
## [91] Hot  Hot  Hot  Hot
## Levels: Cold Hot Toxic
```