

Nikolas David Riapira R.
Andres Felipe Hernandez J.
Santiago Avila Quiroz
Andres Malpica Montenegro
Brayan Stiven Zambrano Sanchez

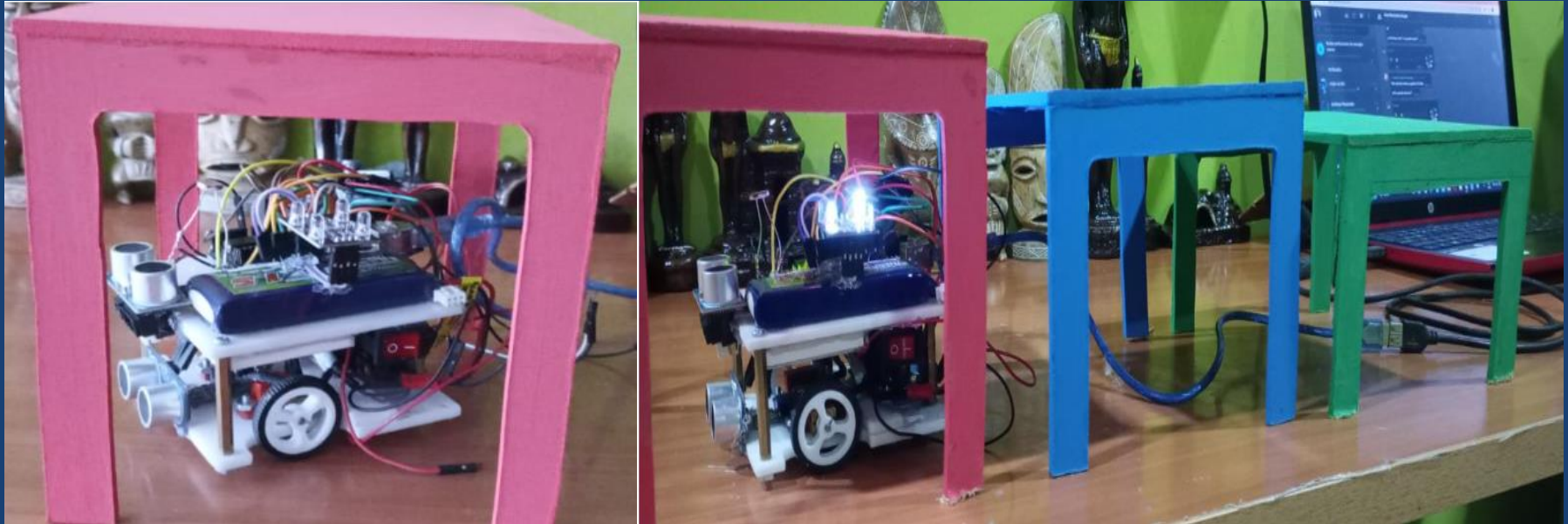
Assignment 3: “Supervised learning final project”

Electiva Área electrónica - 2023



AMBIENTES

Para el robot se tienen los respectivos sensores de distancia, luz y color; esto con respecto a los siguientes ambientes:



AMBIENTES Y PREDICIONES

Teniendo en cuenta los tres ambientes, se utilizan los modelos de predicción correspondientes a:

- 1.) Knn.*
- 2.) Logistic regression.*
- 3.) Decisión Tree.*
- 4.) Random forest.*

DATASET: *Respecto al dataset utilizado al ser varias la forma de censado, los datos se distribuyen en un total de 106 filas de datos obtenidos por los sensores en un espacio de 5 variables, dando un total de aproximadamente de 530 datos; incluyendo a esto una variable de nombre “Ambiente” que nos indica a que ambiente corresponde los datos obtenidos por fila de los sensores . Los nombres de las variables con los datos de sensado son:*

“ultrasónico”, “fotorresistencia” y el sensor de color al necesitar de tres variables para poder hacer la distinción de color se tienen los nombres de “color1”, “color2” y “color3”.

KNN:

```
1 library(tidyverse)
2 library(caret)
3 library(rgmmodels)
4 Dataset1 <- read.csv("D:/Usr/Desktop/Andres/8vo Semestre/DataScience/AssignmentThree/Datasets/DATA_SET_3_SENSORES.csv")
5 Dataset1_df <- Dataset1
6 Dataset1_df$ambiente <- factor(Dataset1_df$ambiente, levels = c("UNO", "DOS", "TRES"))
7 Dataset1_norm <- Dataset1_df
8 #seleccion de variables a normalizar
9 variables <- c("ultrasonico", "fotorresistencia", "color1", "color2", "color3")
10 #normalizacion min-max
11 Dataset1_norm[, variables] <- scale(Dataset1_norm[, variables])
12 #Asignacion niveles
13 Dataset1_norm$ambiente <- factor(Dataset1_norm$ambiente, levels = c("UNO", "DOS", "TRES"))
14
15 # resumen datos normalizados
16 head(Dataset1_norm)
17
18 #index for random sampling
19 sample.indexA <- sample(1:nrow(Dataset1_norm)
20                        ,nrow(Dataset1_norm)*0.7
21                        ,replace = F)
22
23 predictorsA <- c("ultrasonico", "fotorresistencia", "color1", "color2", "color3")
24
25 train.dataA <- Dataset1_norm[sample.indexA
26                             ,c(predictorsA,"ambiente")
27                             ,drop=F]
28
29 test.dataA <- Dataset1_norm[-sample.indexA
30                             ,c(predictorsA,"ambiente")
31                             ,drop=F]
```

Resumidamente la primera sección conduce el dataset a los procesos de:

- Importar el dataset desde la dirección de archivo en el disco.(dirección id).
- Concatenación de factores a la columna de la variable “ambiente”.
- Creación de dataset auxiliares como el Dataset1_norm.
- selección de variables a normalizar, EJ: ultrasónico, fotorresistencia.
- Se asignan los niveles con los elementos como factores de la variable ambiente.
- Se observa o comprueba la normalización de los datos.
- Se asigna el porcentaje de 70% entrenamiento y 30% datos de prueba.

KNN

```
33 ##### Knn + Normalisation (min-max) #####
34 ctrl <- trainControl(method="cv", p=0.7)
35 knnFitAmbiente <- train(ambiente ~ ultrasonico+fotorresistencia+color1+color2+color3
36   , data = train.dataA
37   , method = "knn"
38   , trControl = ctrl
39   , preProcess = c("center","scale")# range = MIN-MAX "center" , "scale" for z-score
40   , tuneLength = 20)
41
42 #Output of knn fit DESDE ACA
43 knnFitAmbiente
44 #plot(knnFitAmbiente)
45
46 #Get predictions for the testing data
47 knnPredict <- predict(knnFitAmbiente, newdata = test.dataA)
48
49 #Get the confusion matrix to see accuracy value and other parameter values
50 confusionMatrix(knnPredict, test.dataA$ambiente)
51
52 CrossTable(test.dataA$ambiente, knnPredict, prop.chisq = FALSE)
53
54
55 saveRDS(knnFitAmbiente, "knn_model.rds")
56
```

En esta segunda sección se encuentran los procesos de:

- *Asignar las variables predictoras a los dataset auxiliares de entrenamiento y prueba.*
- *Aplicar el método de “min.max center scale z-score según especificaciones de la librería caret y variables predictoras.*
- *Obtener los datos salientes para las predicciones con las variables probadas.*
- *Probar la predicción asignándola a una tabla para presentarla.*
- *Salvar el resultado o modelo con los datos escogidos al comienzo asegurando así el modelo.*

KNN

```
1 library(tidyverse)
2 library(caret)
3 library(gmodels)
4 #Load Model
5 NewknnFitAmbiente <- readRDS("knn_model.rds")
6 nuevos_datos1_norm <- PruebaNuevos
7 nuevos_datos1_norm[, variables] <- scale(nuevos_datos1_norm[, variables])
8 nuevos_datos1_norm$ambiente <- factor(nuevos_datos1_norm$ambiente, levels = c("UNO", "DOS", "TRES"))
9
10 #nuevos_datos <- read.csv("D:/Usr/Desktop/Andres/8vo Semestre/DataScience/AssignmentThree/Datasets/PruebaNuevos.csv")
11 predicciones1 <- predict(NewknnFitAmbiente, newdata = nuevos_datos1_norm)
12
13
14 CrossTable(nuevos_datos1_norm$ambiente, predicciones1, prop.chisq = FALSE)
15
```

En esta sección como un agregado al haber guardado el modelo probado anteriormente en esta sección se usan diferentes sentencias de control o asignación de variables para usar el mismo modelo con nuevos datos de prueba.

LOGISTIC REGRESSION

```
1 library(caret)
2 library(tidyverse)
3 library(gmodels)
4
5 Dataset1 <- read.csv("D:/Usr/Desktop/Andres/8vo Semestre/DataScience/AssignmentThree/Datasets/DATA_SET_3_SENSORES.csv")
6 Dataset1_df <- Dataset1
7 Dataset1_df$ambiente <- factor(Dataset1_df$ambiente, levels = c("UNO", "DOS", "TRES"))
8
9 head(Dataset1_norm)
10 sample.indexA2 <- sample(1:nrow(Dataset1_norm)
11                           ,nrow(Dataset1_norm)*0.7
12                           ,replace = F)
13
14 predictorsA2 <- c("ultrasonico", "fotorresistencia", "color1", "color2", "color3")
15
16 train.dataA2 <- Dataset1_norm[sample.indexA2
17                               ,c(predictorsA,"ambiente")
18                               ,drop=F]
19
20 test.dataA2 <- Dataset1_norm[-sample.indexA2
21                              ,c(predictorsA,"ambiente")
22                              ,drop=F]
23
```

- Inicialmente así como en el método de knn se tiene que:
- Importar el dataset desde la dirección de archivo en el disco.(dirección id).
- Concatenación de factores a la columna de la variable “ambiente”.
- Creación de dataset auxiliares como el Dataset1_df.
- Se asigna el porcentaje de 70% entrenamiento y 30% datos de prueba.
- Se especifican las variables predictoras en “predictorsA2”.
- Se asignan los procesos en train y test data según los porcentajes mencionados.

LOGISTIC REGRESSION

```
24 #####
25 fit.control <- trainControl(method = "repeatedcv"
26                             ,number = 10, repeats = 20)
27
28 LRfit <- train(ambiente ~ ultrasonico + fotorresistencia + color1 + color2 + color3
29               ,data = train.dataA2
30               ,method = "multinom"
31               ,trControl = fit.control, trace = FALSE)
32
33 LRfit
34
35 LRPredict <- predict(LRfit, test.dataA2, type = "prob")
36 LRPredict
37 summary(LRfit)
38 |
39 saveRDS(LRfit, "LR_model.rds")
40
```

la segunda sección corresponde a:

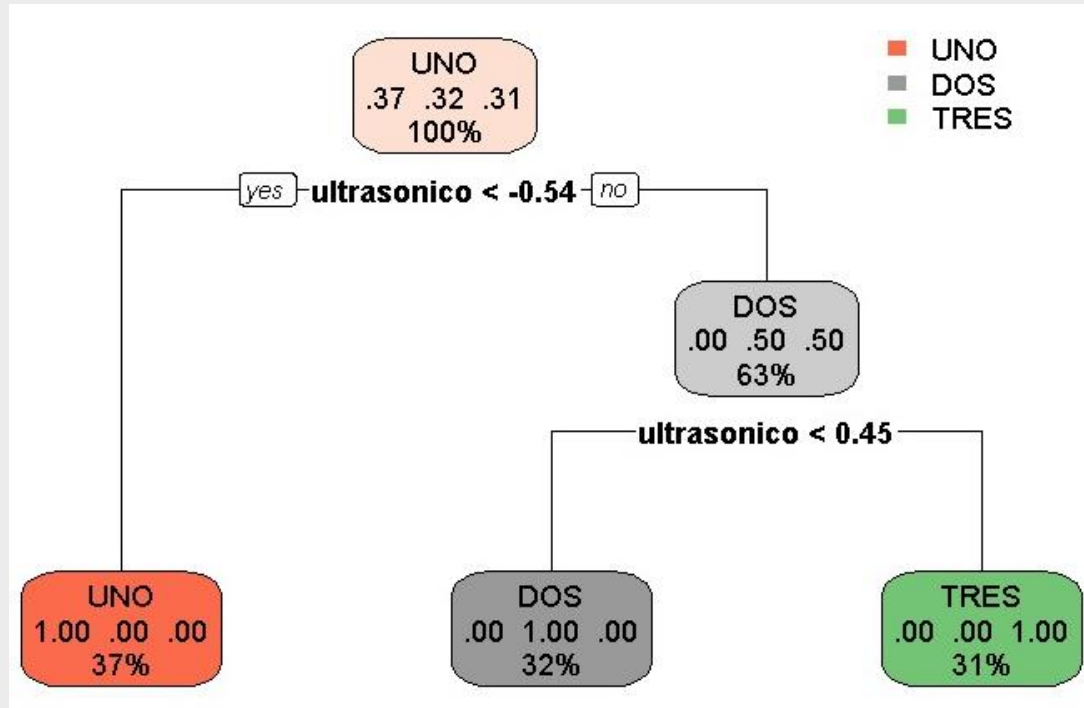
- *La configuración de modelo con método “repeated” configurado a 20 repeticiones.*
- *Se usa train para entrenamiento según el porcentaje escogido de 70%.*
- *Se configura el carácter multinomial del método con las variables predictoras y se hace normalización y predicción*
- *Se guarda el modelo en términos para su posterior normalización y predicción en LR_models.rds con datos de prueba nuevos.*

DECISION TREE

```
1 library(rpart)
2 library(rpart.plot)
3
4 predictorsA3 <- c("ultrasonico", "fotorresistencia", "color1", "color2", "color3")
5 #index for random sampling
6 sample.indexA3 <- sample(1:nrow(Dataset1_norm)
7                           ,nrow(Dataset1_norm)*0.7
8                           ,replace = F)
9
10 train.dataA3 <- Dataset1_norm[sample.indexA3
11                               ,c(predictorsA3,"ambiente")
12                               ,drop=F]
13
14 test.dataA3 <- Dataset1_norm[-sample.indexA3
15                               ,c(predictorsA3,"ambiente")
16                               ,drop=F]
17
18 arbol_ambiente <- rpart(ambiente ~ ultrasonico+fotorresistencia+color1+color2+color3
19                           ,data = train.dataA3)
20
21 rpart.plot(arbol_ambiente, type = 2, extra = 104)
22
23 saveRDS(arbol_ambiente, "DT_model.rds")
24
```

- Concatenación de variables predictoras a predictorsA3.
- Se asigna el porcentaje de 70% entrenamiento y 30% datos de prueba.
- Se usan train.data y test.data para distribución de datos según el porcentaje anterior teniendo en cuenta variables predictoras concatenadas en predictorsA3.
- Se normaliza y predice según los datos dando como resultado el árbol de decisión.

DECISION TREE



- Se consigue el árbol de decisión como lo muestra la imagen siendo que este resumidamente detecta la distribución de datos en la variable de ambiente, al detectar en el sensor ultrasónico un valor menor a -0.54 este se detecta como el ambiente 1, si este es mayor la probabilidad se distribuye en los otros dos ambientes si el dato es mayor o menor a 0.45.

RANDOM FOREST

```
1 library(randomForest)
2
3 data.samples <- sample(1:nrow(Dataset1_norm),
4                       nrow(Dataset1_norm) * 0.7, replace = FALSE)
5
6 training.dataA4 <- Dataset1_norm[data.samples, ]
7
8 test.dataA4 <- Dataset1_norm[-data.samples, ]
9
10 RFfit.rf <- randomForest(ambiente ~ ultrasonico + fotorresistencia + color1 + color2 + color3
11                        ,data = training.dataA4)
12
13 Aprediction.rf <- predict(RFfit.rf, test.dataA4)
14
15 table(test.dataA4$ambiente, Aprediction.rf)
16
17 saveRDS(RFfit.rf, "RF_model.rds")
```

- Se asigna el porcentaje de 70% entrenamiento y 30% datos de prueba.
- Se usan *training.dataA4* y *test.dataA4* para asignar dataset auxiliar y concatenación de variables predictoras con dataset de entrenamiento.
- Se normaliza y predice según los datos dando como resultado la predicción
- Se guarda el modelo para usarlo nuevamente con valores de prueba nuevos.

GRACIAS



HUMANISMO,
CIENCIA Y
TECNOLOGÍA
AL SERVICIO DE LA
SOCIEDAD

   Universidad ECCI
 @UniversidadECCI
 universidad.ecci

info@ecci.edu.co

PBX: (57 1) 353 71 71

Cra. 19 No. 49 - 20

www.ecci.edu.co 