Proyecto Final

Acevedo Mora Andres 55305 Chavez Miguel Angel 80811
Ducuara Quesada Daniela 85742 Garcia Castillo Fenando 61865
Morales Solano Javier Sebastian 73322 Palacios Diego Alejandro 46026

2023-05-29

Abstract

This paper presents an overview of the final project of the elective course Electronics Area I at ECCI University, regarding the training of models for the classification of hostile or difficult to access environments for human personnel. During the training process, certain environments were designed to facilitate the acquisition of data from the sensors used during the development of the project. In this case, a BMP180 barometric pressure sensor was used, which detects both pressure and temperature, and a TCS230 colour sensor, which is capable of identifying the RGB colours in the environment.

Resumen.

En el presente documento se presentará un vistazo al proyecto final de la electiva de Área Electrónica I de la universidad ECCI, con respecto al entrenamiento de modelos de clasificación de ambientes hostiles o de difícil acceso para personal humano. Durante el proceso de entrenamiento, se diseñaron determinados ambientes para facilitar la adquisición de los datos de los sensoores empleados durante el desarrollo del proyecto. En este caso, se empleó un sensor de presión barométrica BMP180, el cual detecta tanto presión como temperatura, y un sensor de color capaz de identificar los porcentajes de color RGB en el ambiente.

Datasets.

The data collected to train the prediction models selected for the final project will be treated to ensure better results during training. The conclusions will address the problems encountered during data collection.

```
df$env <- as.factor(df$env)
df$select._color <- as.factor(df$select._color)
df$temp_C <- as.numeric(as.character(df$temp_C))
df$press <- as.numeric(as.character(df$press))
summary(df)
```

```
    freqRed          freqBlue      select._color     temp_C          press      
 Min.   :  0.00   Min.   : 0.00   blue: 457     Min.   :22.18   Min.   :750.2  
 1st Qu.:  0.00   1st Qu.: 0.00   red : 415     1st Qu.:22.25   1st Qu.:750.5  
 Median :  0.00   Median : 0.00   NA's:1457     Median :22.91   Median :751.2  
 Mean   : 29.83   Mean   :25.07                 Mean   :28.79   Mean   :751.0  
 3rd Qu.: 33.00   3rd Qu.:54.00                 3rd Qu.:41.77   3rd Qu.:751.2  
 Max.   :125.00   Max.   :84.00                 Max.   :45.20   Max.   :751.3  
    env     
 env A:872  
 env B:746  
 env C:711  
```

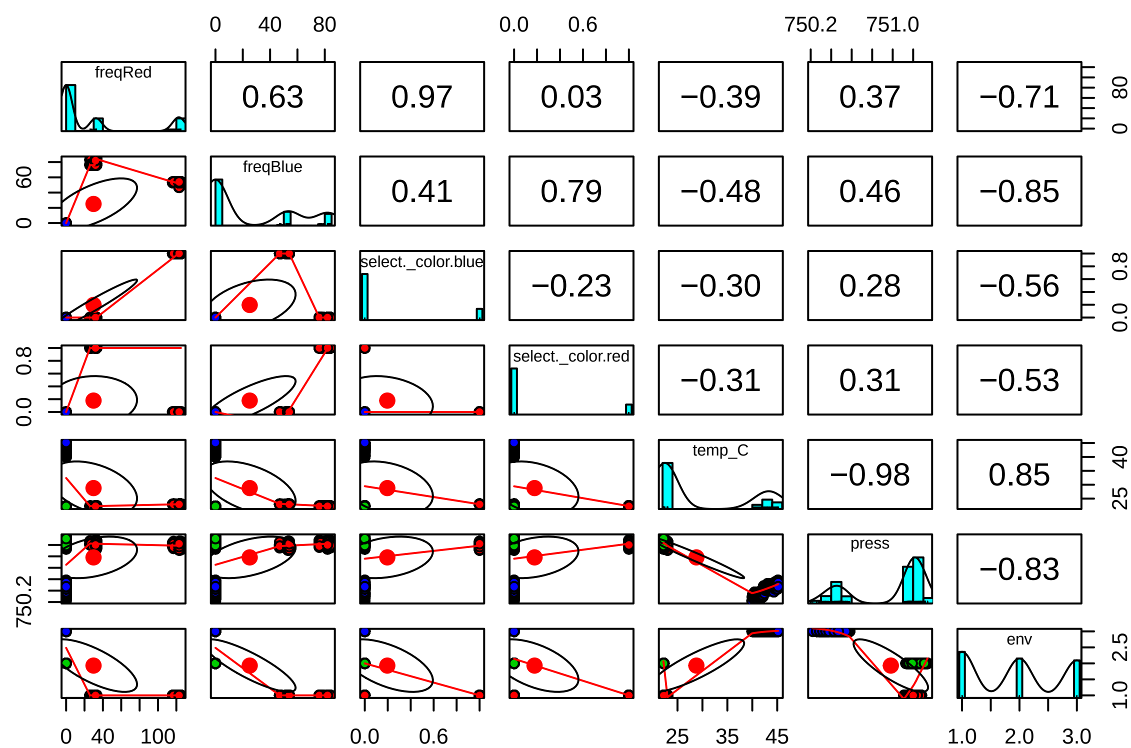
Data processing.

Next, we pre-process the data to ensure that the values in the select_color column are converted to dummy variables for later use in the models.

```
library(caret)
dummies <- dummyVars(~ select._color, data = df, na.action=na.pass)
dummy_data <- predict(dummies, newdata = df)
dummy_data[is.na(dummy_data)] <- 0
df <- cbind(df, dummy_data)
summary(df)
```

```
    freqRed          freqBlue     select._color     temp_C          press
 Min.   :  0.00   Min.   : 0.00   blue: 457     Min.   :22.18   Min.   :750.2
 1st Qu.:  0.00   1st Qu.: 0.00   red : 415     1st Qu.:22.25   1st Qu.:750.5
 Median :  0.00   Median : 0.00   NA's:1457     Median :22.91   Median :751.2
 Mean   : 29.83   Mean   :25.07                 Mean   :28.79   Mean   :751.0
 3rd Qu.: 33.00   3rd Qu.:54.00                 3rd Qu.:41.77   3rd Qu.:751.2
 Max.   :125.00   Max.   :84.00                 Max.   :45.20   Max.   :751.3
    env      select._color.blue select._color.red
 env A:872   Min.   :0.0000     Min.   :0.0000
 env B:746   1st Qu.:0.0000     1st Qu.:0.0000
 env C:711   Median :0.0000     Median :0.0000
             Mean   :0.1962     Mean   :0.1782
             3rd Qu.:0.0000     3rd Qu.:0.0000
             Max.   :1.0000     Max.   :1.0000
```

```
library(psych)
pairs.panels(df[c("freqRed",
                  "freqBlue",
                  "select._color.blue",
                  "select._color.red",
                  "temp_C",
                  "press",
                  "env")],
             pch=21, bg=c("red","green3","blue", "orange")[unclass(df$env)])
```



Separation of data 80-20 for training purposes.

```
df$select._color <- NULL
df.idx <- createDataPartition(df$env, p = 0.8, list = FALSE)
df.train <- df[df.idx,]
df.test <- df[-df.idx,]
```

Knn Model.

During data preparation, parameters associated with training, such as cross-validation and pre-processing centering and scaling, are configured to ensure that the data is on a smaller scale. This can ensure a smaller difference in data scale.

```
k.grid <- expand.grid(k = seq(1, 20, by = 2))
control.knn <- trainControl(method = 'cv', number = 15)
model.knn <- train(
    env ~ .,
    data = df.train,
    method = 'knn',
    preProcess = c('center', 'scale'),
    trControl = control.knn,
    tuneGrid = k.grid
    )
model.knn
```

```
## k-Nearest Neighbors
##
## 1864 samples
```

```
##    6 predictor
##    3 classes: 'env A', 'env B', 'env C'
##
## Pre-processing: centered (6), scaled (6)
## Resampling: Cross-Validated (15 fold)
## Summary of sample sizes: 1740, 1740, 1739, 1740, 1742, 1739, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy  Kappa
##    1  1         1
##    3  1         1
##    5  1         1
##    7  1         1
##    9  1         1
##   11  1         1
##   13  1         1
##   15  1         1
##   17  1         1
##   19  1         1
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 19.
```

Model Test

The appropriate tests are run on 30% of the data to check the accuracy of the model and a report is generated with the prediction model.

```
predicted.knn <- predict(model.knn, newdata = df.test)
conf.matrix <- confusionMatrix(predicted.knn, df.test$env)
conf.matrix$table
```

```
##           Reference
## Prediction env A env B env C
##      env A   174     0     0
##      env B     0   149     0
##      env C     0     0   142
```

```
conf.matrix$overall["Accuracy"]
```

```
## Accuracy
##        1
```

Without further training data and/or further test data, it can be concluded that the KNN model is over-fitting and therefore 100% accuracy is achieved.

Multinominal logitic regression

Considering the results obtained when training the kNN model, a multinomial logistic regression model approximation is chosen because it is necessary to ensure that the model does not overfit in order to guarantee the credibility of the model.

```
library(tidyverse)
library(dplyr)
library(MASS)
library(nnet)
control.multinom <- trainControl(method = 'cv', number = 10)
```

```r
tune.grid <- expand.grid(decay = seq(0, 1, by = 0.1))
model.multinom <- multinom (
    env ~ temp_C + press + freqBlue + freqRed,
    data = df.train,
    dacay = 0.1,
    iter = 500
)
```

```
## # weights:  18 (10 variable)
## initial  value 2047.813306
## final  value 0.000000
## converged
```

```r
model.multinom
```

```
## Call:
## multinom(formula = env ~ temp_C + press + freqBlue + freqRed,
##     data = df.train, dacay = 0.1, iter = 500)
##
## Coefficients:
##       (Intercept)    temp_C    press  freqBlue   freqRed
## env B 0.002740140 -7.351523 2.369848 -21.98399 -25.30962
## env C 0.002819978 14.396915 1.557764 -25.24160 -29.05955
##
## Residual Deviance: 0
## AIC: 20
```

testing Multinominal logitic regression.

```r
# Make predictions on test data
predicted.multinom <- predict(model.multinom, newdata = df.test)

# Evaluate model performance
conf.matrix <- confusionMatrix(predicted.multinom, df.test$env)
conf.matrix$table
```

```
##           Reference
## Prediction env A env B env C
##      env A   174     0     0
##      env B     0   149     0
##      env C     0     0   142
```

```r
conf.matrix$overall["Accuracy"]
```

```
## Accuracy
##        1
```

Several points can be considered regarding the MLR model. However, it cannot be ruled out that the training dataset has unwanted parameters or does not capture enough information. There are several points to consider, such as the fact that the colour sensor has been found to have errors in capturing data in certain environments.

Random Forest

Considering the kNN and multinomial models above, the Random Forest model is chosen. In this case, by specifying the predictors to be used and the target, a forest of 500 trees is generated to ensure a more accurate prediction without pronounced overfitting.

```
library(randomForest)
df$env = factor(df$env)
rf <-df[complete.cases(df),]
df.idx.rf<-createDataPartition(df$env,p=0.7,list = F)

model.rf <- randomForest(
    x = df[df.idx.rf, -which(names(df) == "env")],
    y = df[df.idx.rf, "env"],
    ntree = 500,
    keep.forest = TRUE
)
summary(model.rf)
```

```
##                 Length Class  Mode
## call               5   -none- call
## type               1   -none- character
## predicted       1632   factor numeric
## err.rate        2000   -none- numeric
## confusion         12   -none- numeric
## votes           4896   matrix numeric
## oob.times       1632   -none- numeric
## classes            3   -none- character
## importance         6   -none- numeric
## importanceSD       0   -none- NULL
## localImportance    0   -none- NULL
## proximity          0   -none- NULL
## ntree              1   -none- numeric
## mtry               1   -none- numeric
## forest            14   -none- list
## y               1632   factor numeric
## test               0   -none- NULL
## inbag              0   -none- NULL
```

testing Random Forest.

```
rf.predict <- predict(model.rf, newdata = df.test)
summary(rf.predict)
```

```
## env A env B env C
##   174   149   142
```

save models

```
saveRDS(model.knn, paste0(parentFolder,"/models/knnModel.rds"))
saveRDS(model.multinom, paste0(parentFolder,"/models/multinom.rds"))
saveRDS(model.rf, paste0(parentFolder,"/models/rf_model.rds"))
```

Conclusions

It can be seen that there may very well have been problems with the acquisition of the data from the colour sensor, which shows a number of errors that may have affected the training of the models, causing both bias and overfitting. In this case, overfitting may well have occurred. However, the possible replacement of the sensors used during the development of the final project by a BME280, which has the possibility of using 3 predictors, which could have guaranteed the non-proliferation of problems during the development of the final project, is taken into account belatedly.

Repository GITHUB