Classificação de Veículos

Luan Roger Santos Santana

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

Preparing data

Loading Data

```
data_raw <- read.csv("../data_sets/Material 03 - 11 - Banco - Dados.csv")
data_raw_new_cases <- read.csv("../data_sets/Material 03 - 11 - Banco - Dados - Novos Casos.csv")
```

Cleaning data

```
data <- data_raw
data_new_cases <- data_raw_new_cases
print(head(data))
```

```
##   age        job marital education default balance housing loan  y
## 1  30 unemployed married   primary      no    1787      no   no no
## 2  33   services married secondary      no    4789     yes  yes no
## 3  35 management  single  tertiary      no    1350     yes   no no
## 4  30 management married  tertiary      no    1476     yes  yes no
## 5  59 bluecollar married secondary      no       0     yes   no no
## 6  35 management  single  tertiary      no     747      no   no no
```

```
print(head(data_new_cases))
```

```
##   age        job marital education default balance housing loan y
## 1  60 unemployed married   primary      no    2000     yes  yes ?
## 2  33   services married secondary     yes    3000     yes   no ?
## 3  15 management  single  tertiary      no    1350     yes   no ?
```

Creating data partitioning

```
set.seed(1988)
ran <- sample(1:nrow(data), 0.8 * nrow(data))
training_data <- data[ran,]
test_data <- data[-ran,]
```

Training

Using KNN

Creating the model

```
tuneGrid <- expand.grid(k = c(1,3,5,7,9))
set.seed(1988)
knn <- train(y ~ ., data = training_data, method = "knn", tuneGrid=tuneGrid)
print(knn)
```

```
## k-Nearest Neighbors
##
## 240 samples
##   8 predictor
##   2 classes: 'no', 'yes'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 240, 240, 240, 240, 240, 240, ...
## Resampling results across tuning parameters:
##
##   k  Accuracy   Kappa
##   1  0.7977511  0.18639365
##   3  0.8023338  0.10034546
##   5  0.8187057  0.05121811
##   7  0.8347417  0.07011804
##   9  0.8418864  0.04785112
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 9.
```

```
prediction.knn <- predict(knn, test_data)
cf_matrix <- confusionMatrix(prediction.knn, as.factor(test_data$y))
print(cf_matrix)
```

Checking the model with training data

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction no yes
##        no  54   6
##        yes  0   0
##
##                Accuracy : 0.9
##                  95% CI : (0.7949, 0.9624)
##     No Information Rate : 0.9
##     P-Value [Acc > NIR] : 0.60645
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : 0.04123
##
```

```
##             Sensitivity : 1.0
##             Specificity : 0.0
##          Pos Pred Value : 0.9
##          Neg Pred Value : NaN
##              Prevalence : 0.9
##          Detection Rate : 0.9
##    Detection Prevalence : 1.0
##       Balanced Accuracy : 0.5
##
##        'Positive' Class : no
##
```

Checking for new cases

```
prediction.knn_new_data <- predict(knn, data_new_cases)
data_new_cases$y <- NULL
result <- cbind(data_new_cases, tipo=prediction.knn_new_data)
print(result)
```

```
##   age        job marital education default balance housing loan tipo
## 1  60 unemployed married   primary      no    2000     yes  yes   no
## 2  33   services married secondary     yes    3000     yes   no   no
## 3  15 management  single  tertiary      no    1350     yes   no   no
```