RandallPlylerCh7

Randall Plyler

2/11/2022

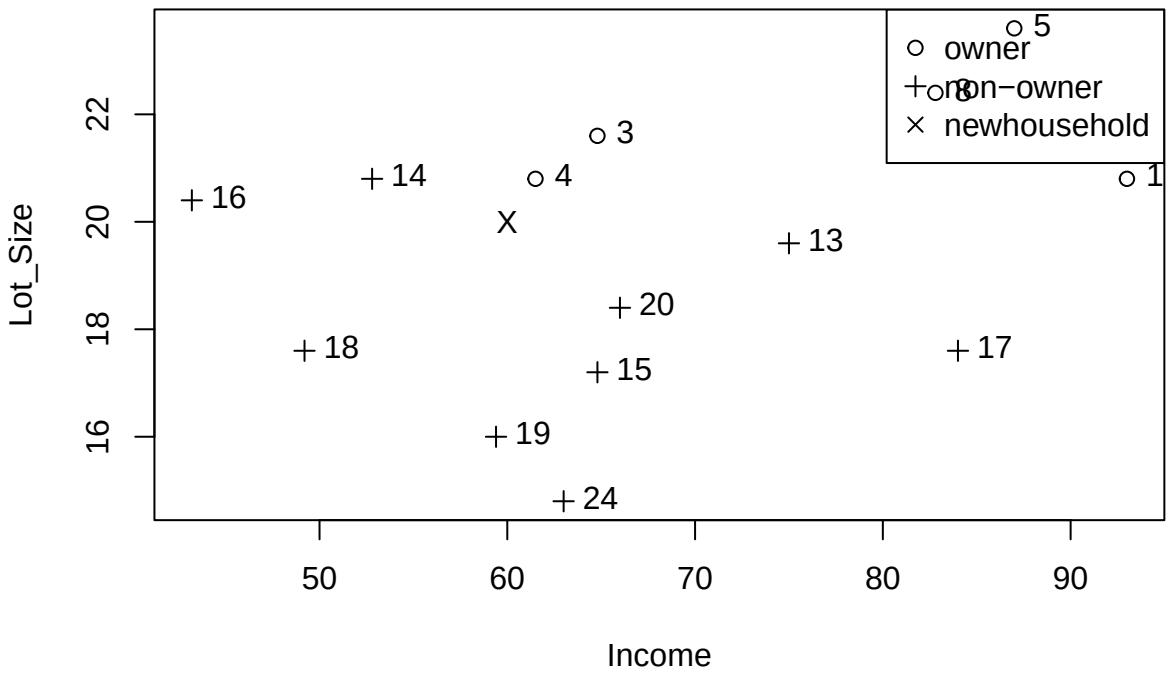
```
library(FNN)
library(caret)
```

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

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

```
mower.df <- read.csv("C:/Users/randa/Dropbox/Masters/Winter/TBANLT 560 Data Mining/Files/DMBA-R-datasets
set.seed(111)
train.index <- sample(row.names(mower.df), 0.6*dim(mower.df)[1])
valid.index <- setdiff(row.names(mower.df), train.index)
train.df <- mower.df[train.index, ]
valid.df <- mower.df[valid.index, ]
## new household
new.df <- data.frame(Income = 60, Lot_Size = 20)
```

```
## scatter plot
plot(Lot_Size ~ Income, data=train.df, pch=ifelse(train.df$Ownership=="Owner", 1, 3))
text(train.df$Income, train.df$Lot_Size, rownames(train.df), pos=4)
text(60, 20, "X")
legend("topright", c("owner", "non-owner", "newhousehold"), pch = c(1, 3, 4))
```




```
# initialize normalized training, validation data, complete data frames to originals
train.norm.df <- train.df
valid.norm.df <- valid.df
mower.norm.df <- mower.df
# use preProcess() from the caret package to normalize Income and Lot_Size.
norm.values <- preProcess(train.df[, 1:2], method=c("center", "scale"))
train.norm.df[, 1:2] <- predict(norm.values, train.df[, 1:2])
valid.norm.df[, 1:2] <- predict(norm.values, valid.df[, 1:2])
mower.norm.df[, 1:2] <- predict(norm.values, mower.df[, 1:2])
new.norm.df <- predict(norm.values, new.df)
```

```
# initialize a data frame with two columns: k, and accuracy.
accuracy.df <- data.frame(k = seq(1, 14, 1), accuracy = rep(0, 14))
# compute knn for different k on validation.
for(i in 1:14) {
  knn.pred <- knn(train.norm.df[, 1:2], valid.norm.df[, 1:2],  cl = train.norm.df[, 3], k = i)
  accuracy.df[i, 2] <- confusionMatrix(as.factor(knn.pred), as.factor(valid.norm.df[, 3]))$overall[1]
}
```

```
## Warning in confusionMatrix.default(as.factor(knn.pred),
## as.factor(valid.norm.df[, : Levels are not in the same order for reference and
## data. Refactoring data to match.

## Warning in confusionMatrix.default(as.factor(knn.pred),
## as.factor(valid.norm.df[, : Levels are not in the same order for reference and
```

```
## data. Refactoring data to match.

## Warning in confusionMatrix.default(as.factor(knn.pred),
## as.factor(valid.norm.df[, : Levels are not in the same order for reference and
## data. Refactoring data to match.

## Warning in confusionMatrix.default(as.factor(knn.pred),
## as.factor(valid.norm.df[, : Levels are not in the same order for reference and
## data. Refactoring data to match.

## Warning in confusionMatrix.default(as.factor(knn.pred),
## as.factor(valid.norm.df[, : Levels are not in the same order for reference and
## data. Refactoring data to match.

## Warning in confusionMatrix.default(as.factor(knn.pred),
## as.factor(valid.norm.df[, : Levels are not in the same order for reference and
## data. Refactoring data to match.

## Warning in confusionMatrix.default(as.factor(knn.pred),
## as.factor(valid.norm.df[, : Levels are not in the same order for reference and
## data. Refactoring data to match.
```

```r
accuracy.df
```

```
##     k accuracy
## 1   1      0.4
## 2   2      0.3
## 3   3      0.5
## 4   4      0.4
## 5   5      0.5
## 6   6      0.3
## 7   7      0.4
## 8   8      0.4
## 9   9      0.5
## 10 10      0.3
## 11 11      0.3
## 12 12      0.3
## 13 13      0.3
## 14 14      0.3
```

```r
nn <- FNN::knn(train = train.norm.df[, 1:2], test = new.norm.df,
          cl = train.norm.df[, 3], k = 3)
row.names(train.df)[attr(nn, "nn.index")]
```

```
## [1] "4"  "14" "3"
```

```r
accuracy.df[i, 2] <- confusionMatrix(as.factor(knn.pred), as.factor(valid.norm.df[, 3]))$overall[1]
```

```
## Warning in confusionMatrix.default(as.factor(knn.pred),
## as.factor(valid.norm.df[, : Levels are not in the same order for reference and
## data. Refactoring data to match.
```

```r
### Table 7.3

# initialize a data frame with two columns: k, and accuracy.
accuracy.df <- data.frame(k = seq(1, 14, 1), accuracy = rep(0, 14))
# compute knn for different k on validation.
for(i in 1:14) {
  knn.pred <- knn(train.norm.df[, 1:2], valid.norm.df[, 1:2],  cl = train.norm.df[, 3], k = i)
  accuracy.df[i, 2] <- confusionMatrix(as.factor(knn.pred), as.factor(valid.norm.df[, 3]))$overall[1]
}
```

```
## Warning in confusionMatrix.default(as.factor(knn.pred),
## as.factor(valid.norm.df[, : Levels are not in the same order for reference and
## data. Refactoring data to match.

## Warning in confusionMatrix.default(as.factor(knn.pred),
## as.factor(valid.norm.df[, : Levels are not in the same order for reference and
## data. Refactoring data to match.

## Warning in confusionMatrix.default(as.factor(knn.pred),
## as.factor(valid.norm.df[, : Levels are not in the same order for reference and
## data. Refactoring data to match.

## Warning in confusionMatrix.default(as.factor(knn.pred),
## as.factor(valid.norm.df[, : Levels are not in the same order for reference and
## data. Refactoring data to match.

## Warning in confusionMatrix.default(as.factor(knn.pred),
## as.factor(valid.norm.df[, : Levels are not in the same order for reference and
## data. Refactoring data to match.

## Warning in confusionMatrix.default(as.factor(knn.pred),
## as.factor(valid.norm.df[, : Levels are not in the same order for reference and
## data. Refactoring data to match.

## Warning in confusionMatrix.default(as.factor(knn.pred),
## as.factor(valid.norm.df[, : Levels are not in the same order for reference and
## data. Refactoring data to match.
```

```r
accuracy.df# use knn() to compute knn.
```

```
##     k accuracy
## 1   1      0.4
## 2   2      0.3
## 3   3      0.5
## 4   4      0.4
## 5   5      0.5
## 6   6      0.3
## 7   7      0.4
## 8   8      0.4
## 9   9      0.5
## 10 10      0.3
## 11 11      0.3
## 12 12      0.3
```

```
## 13 13      0.3
## 14 14      0.3
```

```
# knn() is available in library FNN (provides a list of the nearest neighbors)
# and library class (allows a numerical output variable).

nn <- FNN::knn(train = train.norm.df[, 1:2], test = new.norm.df,
          cl = train.norm.df[, 3], k = 3)
row.names(train.df)[attr(nn, "nn.index")]
```

```
## [1] "4"  "14" "3"
```

```
accuracy.df[i, 2] <- confusionMatrix(as.factor(knn.pred), as.factor(valid.norm.df[, 3]))$overall[1]
```

```
## Warning in confusionMatrix.default(as.factor(knn.pred),
## as.factor(valid.norm.df[, : Levels are not in the same order for reference and
## data. Refactoring data to match.
```

```
#### Table 7.4

knn.pred.new <- knn(mower.norm.df[, 1:2], new.norm.df,
                    cl = mower.norm.df[, 3], k = 4)
row.names(train.df)[attr(nn, "nn.index")]
```

```
## [1] "4"  "14" "3"
```