Chapter 17 - Use Case

Ram Gopal, Dan Philps, and Tillman Weyde

2022

Contents

```
library(caret)
```

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

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

```
library(rattle)
```

```
## Loading required package: tibble
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```r
library(stringr)
```

Use Case - Decision Trees to Model App Ratings

This is likely to be the most complicated use case thus far in the book. Students may struggle a bit, but it provides opportunities to extend this to an involved assignment or a course project. It is useful to impress on the students the importance of data preparation. This use case provides a nice "real-world" data set that will consume significant effort to get the data ready for analysis.

Read the data

- Remind students to read the textual data as strings.

```r
df <- read.csv("C:/Users/rgopal/Google Drive/Maya R/googleplaystore.csv",stringsAsFactors = T)
```

*Quick peek of the data. Illustrates that a number of inherently numeric variables have been read as categorical which need to be fixed.

```r
str(df)
```

```
## 'data.frame':    10841 obs. of  13 variables:
##  $ App           : Factor w/ 9660 levels "\"i DT\" FÃºtbol. Todos Somos TÃ©cnicos.",..: 7229 2563 89
##  $ Category      : Factor w/ 34 levels "1.9","ART_AND_DESIGN",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Rating        : num  4.1 3.9 4.7 4.5 4.3 4.4 3.8 4.1 4.4 4.7 ...
##  $ Reviews       : Factor w/ 6002 levels "0","1","10","100",..: 1183 5924 5681 1947 5924 1310 1464 3
##  $ Size          : Factor w/ 462 levels "1,000+","1.0M",..: 55 30 368 102 64 222 55 118 146 120 ...
##  $ Installs      : Factor w/ 22 levels "0","0+","1,000,000,000+",..: 8 20 13 16 11 17 17 4 4 8 ...
##  $ Type          : Factor w/ 4 levels "0","Free","NaN",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Price         : Factor w/ 93 levels "$0.99 ","$1.00 ",..: 92 92 92 92 92 92 92 92 92 92 ...
##  $ Content.Rating: Factor w/ 7 levels "","Adults only 18+",..: 3 3 3 6 3 3 3 3 3 3 ...
##  $ Genres        : Factor w/ 120 levels "11-Feb-18","Action",..: 11 14 11 11 13 11 11 11 11 13 ...
##  $ Last.Updated  : Factor w/ 1378 levels "1-Apr-16","1-Apr-17",..: 1268 304 7 1318 585 861 830 263 6
##  $ Current.Ver   : Factor w/ 2786 levels "","0.0.0.2","0.0.1",..: 117 998 460 2779 274 110 274 2347
##  $ Android.Ver   : Factor w/ 35 levels "","1.0 and up",..: 17 17 17 20 22 10 17 20 12 17 ...
```

Data Wrangling

Convert Price to numeric

- This is a good opportunity to discuss string manipulation with stringr package.

```r
df$Price=str_replace(df$Price,",","")
df$Price = str_replace(df$Price,"\\$","")
df$Price = as.numeric(df$Price)
```

```
## Warning: NAs introduced by coercion
```

```
df$Price = as.numeric(str_replace_na(df$Price,0))
```

Convert Rating

- To a 3 level factor variable.

```
x = ifelse(is.nan(df$Rating),0,df$Rating)
brakepoints = quantile(x,probs= c(0,0.33,0.67,1))
x = cut(x,breaks = brakepoints,labels = c("lowest","middle","highest"))
x = ifelse(is.na(x),"1",x)
df$Rating = x
```

Convert Reviews, Installs, and Size to numeric

```
df$Reviews = as.numeric(df$Reviews)
df$Installs = as.numeric(df$Installs)
df$Size = as.numeric(df$Size)
```

Remove extraneous variables unlikely to be useful for prediction

```
df = df[-c(1,11,12,13)]
str(df)
```

```
## 'data.frame':    10841 obs. of  9 variables:
##  $ Category      : Factor w/ 34 levels "1.9","ART_AND_DESIGN",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Rating        : chr  "2" "1" "3" "3" ...
##  $ Reviews       : num  1183 5924 5681 1947 5924 ...
##  $ Size          : num  55 30 368 102 64 222 55 118 146 120 ...
##  $ Installs      : num  8 20 13 16 11 17 17 4 4 8 ...
##  $ Type          : Factor w/ 4 levels "0","Free","NaN",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Price         : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Content.Rating: Factor w/ 7 levels "","Adults only 18+",..: 3 3 3 6 3 3 3 3 3 3 ...
##  $ Genres        : Factor w/ 120 levels "11-Feb-18","Action",..: 11 14 11 11 13 11 11 11 11 13 ...
```

Final check for missing values

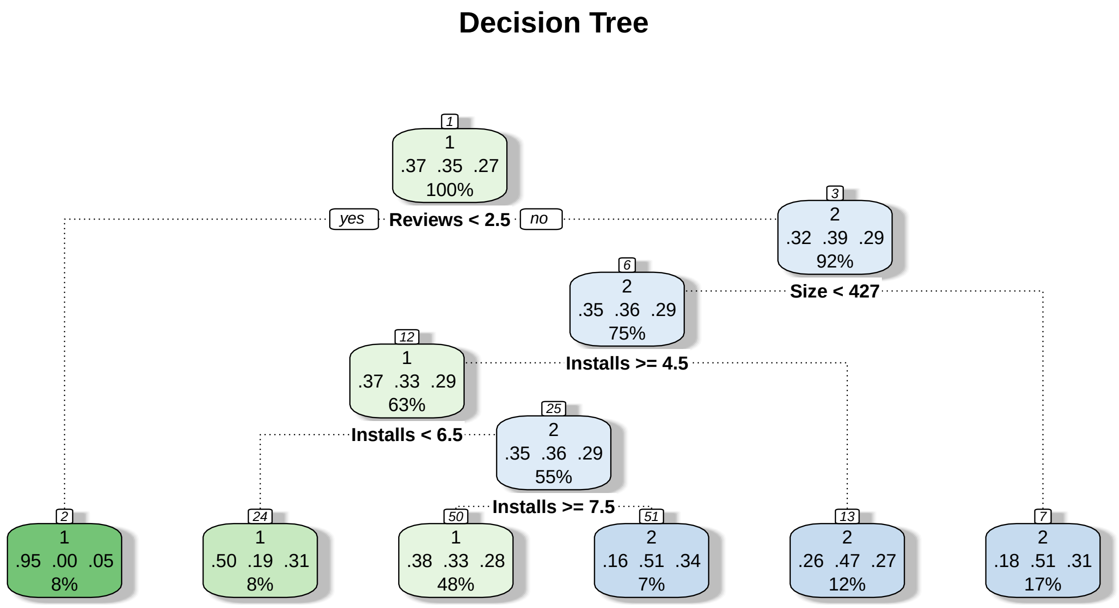
```
table(complete.cases(df))
```

```
##
##  TRUE
## 10841
```

Building a Decision Tree model

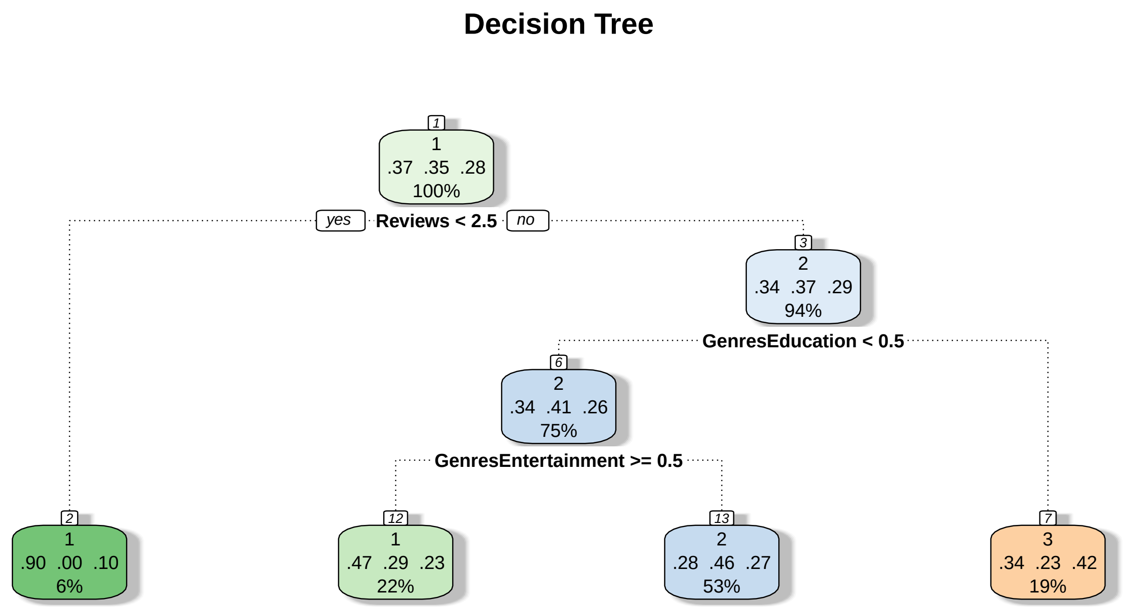
- fancyRpartPlot function from the rattle package is a good one to draw the tree.

```
library(caret)
library(rpart)
dtmodel <- train(Rating ~ ., method = "rpart", data = df,
 trControl=trainControl(method = 'cv',number = 20))
fancyRpartPlot(dtmodel$finalModel,main = "Decision Tree",sub = "")
```



Building a Decision Tree model for the FAMILY Category

```
dtmodel_family <- train(Rating ~ ., method = "rpart", data = df[df$Category=="FAMILY",],
 trControl=trainControl(method = 'cv',number = 20))
fancyRpartPlot(dtmodel_family$finalModel,main = "Decision Tree",sub = "")
```



Assessing Accuracy and Stability of the model

- Overall accuracy

```
mean(dtmodel$resample$Accuracy)
```

```
## [1] 0.4709856
```

- Performance across two validation setups.

```
dtmodel <- train(Rating ~ ., method = "rpart", data = df,
 trControl=trainControl(method = 'cv',number = 10))
paste("Mean of accuracy for 10 fold validation = ",mean(dtmodel$results$Accuracy))
```

```
## [1] "Mean of accuracy for 10 fold validation =  0.436397401994368"
```

```
paste("sd of accuracy for 10 fold validation = ",sd(dtmodel$results$Accuracy))
```

```
## [1] "sd of accuracy for 10 fold validation =  0.0391746858854234"
```

```r
dtmodel <- train(Rating ~ ., method = "rpart", data = df,
 trControl=trainControl(method = 'cv',number = 20))
paste("Mean of accuracy for 20 fold validation = ",mean(dtmodel$results$Accuracy))
```

```
## [1] "Mean of accuracy for 20 fold validation =  0.440300198967615"
```

```r
paste("sd of accuracy for 20 fold validation = ",sd(dtmodel$results$Accuracy))
```

```
## [1] "sd of accuracy for 20 fold validation =  0.0367112242218669"
```

Remarks

There are a number of opportunities to extend the use case. Various models can be assessed for model selection and other approaches to data wrangling can be explored as well.