Appendix A - PDA and Data Cleaning

Ben Kelley

2024-06-30

```r
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.3.3
```

```r
library(reticulate)
```

```
## Warning: package 'reticulate' was built under R version 4.3.3
```

```r
library(tensorflow)
```

```
## Warning: package 'tensorflow' was built under R version 4.3.3
```

```r
library(keras)
```

```
## Warning: package 'keras' was built under R version 4.3.3
```

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.3

## Loading required package: ggplot2

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:tensorflow':
##
##     train
```

```r
library(ROCR)
```

```
## Warning: package 'ROCR' was built under R version 4.3.3
```

```r
use_virtualenv("my_tf_workspace", required = TRUE)

data <- read_csv("~/Data Science Masters Program/DSE6211/project_data.csv")
```

```
## Rows: 36238 Columns: 17
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (5): Booking_ID, type_of_meal_plan, room_type_reserved, market_segment...
## dbl  (11): no_of_adults, no_of_children, no_of_weekend_nights, no_of_week_ni...
## date  (1): arrival_date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# adding a column to the data set which I will assign a 1 or 0 depending on
# customer status
# data[ , 'cancel'] <- NA
# I decided to simply replace the original column entries with 1 and 0 instead
# of creating a new column - I may change this later.

# using if else statement to assign a 1 or 0 to the 'booking_status' column
# denoting 1 for a cancelled booking and 0 for fulfilled booking.
data$booking_status <- ifelse(data$booking_status %in%
                                c('canceled'), 1, 0)

set.seed(123) # setting the seed for reproducibility

training_ind <- createDataPartition(data$booking_status,
                                    p = 0.75,
                                    list = FALSE,
                                    times = 1)
#creating training and test sets
training_set <- data[training_ind, ]
test_set <- data[-training_ind, ]
unique(training_set$type_of_meal_plan)
```

```
## [1] "meal_plan_1"  "not_selected" "meal_plan_2"  "meal_plan_3"
```

```r
unique(training_set$room_type_reserved)
```

```
## [1] "room_type1" "room_type4" "room_type2" "room_type6" "room_type5"
## [6] "room_type7" "room_type3"
```

```r
unique(training_set$arrival_date)
```

```
##   [1] "2017-10-02" "2018-11-06" "2018-02-28" "2018-05-20" "2018-04-11"
##   [6] "2018-09-13" "2017-10-15" "2018-12-26" "2018-07-06" "2018-10-18"
##  [11] "2018-09-11" "2018-06-15" "2017-10-05" "2017-08-10" "2017-10-30"
##  [16] "2017-10-04" "2018-11-25" "2018-04-28" "2017-09-21" "2018-05-19"
##  [21] "2017-09-17" "2017-09-19" "2018-11-13" "2018-12-07" "2018-01-09"
##  [26] "2018-10-07" "2018-04-27" "2018-06-19" "2017-10-17" "2018-11-19"
##  [31] "2018-07-30" "2018-11-01" "2018-06-21" "2018-04-10" "2018-06-27"
##  [36] "2017-11-18" "2017-11-20" "2018-04-06" "2018-12-29" "2018-05-30"
##  [41] "2018-04-22" "2017-11-11" "2018-06-13" "2018-07-28" "2018-04-20"
##  [46] "2018-06-24" "2017-09-10" "2018-12-18" "2018-10-05" "2018-06-03"
##  [51] "2018-04-25" "2018-08-29" "2017-10-25" "2018-03-09" "2018-12-31"
##  [56] "2018-06-28" "2018-04-01" "2018-03-14" "2018-03-04" "2018-10-17"
##  [61] "2018-09-04" "2018-08-09" "2018-05-05" "2018-04-09" "2018-08-30"
##  [66] "2018-08-18" "2018-04-13" "2018-11-17" "2018-12-22" "2018-05-13"
##  [71] "2018-12-13" "2017-10-31" "2018-02-16" "2017-11-02" "2018-09-30"
##  [76] "2018-06-16" "2018-06-26" "2018-11-12" "2018-04-29" "2017-11-14"
##  [81] "2018-09-16" "2018-03-10" "2018-04-24" "2018-03-30" "2018-11-03"
##  [86] "2018-10-14" "2018-06-05" "2018-07-13" "2018-10-20" "2017-10-19"
##  [91] "2018-03-29" "2017-10-09" "2018-07-25" "2018-05-22" "2018-04-15"
##  [96] "2018-06-17" "2018-11-18" "2018-06-01" "2018-05-04" "2018-06-08"
## [101] "2018-04-07" "2018-09-29" "2018-04-04" "2017-07-17" "2018-04-17"
## [106] "2018-01-02" "2018-11-16" "2018-09-19" "2018-09-15" "2018-12-16"
## [111] "2018-06-30" "2018-11-05" "2018-08-17" "2017-10-24" "2018-07-21"
## [116] "2017-09-09" "2018-03-31" "2018-04-02" "2018-08-22" "2018-08-08"
## [121] "2018-08-15" "2018-08-19" "2018-10-10" "2018-03-25" "2018-08-03"
## [126] "2017-09-11" "2018-03-23" "2018-09-26" "2017-07-16" "2018-09-08"
## [131] "2018-12-03" "2018-02-27" "2017-09-22" "2018-08-13" "2018-12-05"
## [136] "2018-03-20" "2018-10-13" "2018-03-01" "2018-04-14" "2018-05-27"
## [141] "2018-03-18" "2018-02-05" "2018-05-12" "2018-10-28" "2017-08-14"
## [146] "2018-09-14" "2018-03-24" "2018-05-31" "2017-11-09" "2018-07-26"
## [151] "2018-05-21" "2018-09-07" "2017-11-15" "2017-09-16" "2017-10-16"
## [156] "2018-07-23" "2018-01-16" "2018-10-15" "2017-09-02" "2017-08-12"
## [161] "2017-08-23" "2018-03-11" "2018-08-28" "2018-12-09" "2018-03-05"
## [166] "2017-08-18" "2018-04-21" "2018-06-18" "2018-05-09" "2018-08-10"
## [171] "2017-08-08" "2018-05-01" "2018-04-03" "2018-05-16" "2018-02-20"
## [176] "2017-12-29" "2017-09-30" "2018-08-01" "2017-12-26" "2018-07-09"
## [181] "2018-06-06" "2018-07-17" "2018-02-22" "2018-07-08" "2018-11-11"
## [186] "2017-10-06" "2017-08-11" "2017-12-15" "2017-10-23" "2017-10-10"
## [191] "2017-09-08" "2018-12-21" "2018-11-10" "2018-08-24" "2018-02-24"
## [196] "2018-07-01" "2018-07-05" "2018-07-11" "2018-10-22" "2018-09-18"
## [201] "2018-10-23" "2017-09-25" "2018-12-10" "2018-10-04" "2018-10-27"
## [206] "2018-03-03" "2018-12-02" "2018-05-14" "2017-09-15" "2018-02-19"
## [211] "2018-04-30" "2018-08-12" "2018-09-09" "2018-01-29" "2018-04-08"
## [216] "2018-08-06" "2018-11-04" "2018-05-29" "2018-01-20" "2017-12-07"
## [221] "2018-11-14" "2018-07-02" "2018-09-27" "2018-09-03" "2017-10-20"
## [226] "2018-09-06" "2018-05-11" "2017-10-13" "2018-10-16" "2017-09-18"
## [231] "2017-08-26" "2018-12-08" "2017-12-27" "2018-06-12" "2018-10-11"
## [236] "2017-10-22" "2018-03-08" "2018-06-07" "2018-04-23" "2018-10-21"
## [241] "2018-08-25" "2017-12-05" "2017-12-24" "2018-06-14" "2018-02-04"
## [246] "2017-12-17" "2018-06-10" "2018-08-21" "2017-10-07" "2018-11-02"
## [251] "2018-03-19" "2018-09-05" "2018-02-03" "2018-04-26" "2018-03-16"
## [256] "2018-10-19" "2017-12-30" "2018-10-03" "2018-06-25" "2017-07-01"
## [261] "2018-06-11" "2018-09-02" "2018-12-30" "2018-01-06" "2018-06-29"
## [266] "2017-07-27" "2018-06-20" "2018-02-26" "2018-07-14" "2018-12-28"
## [271] "2018-07-27" "2018-10-02" "2017-09-05" "2017-09-07" "2018-07-10"
## [276] "2017-08-25" "2017-10-14" "2018-10-01" "2018-04-05" "2018-10-06"
## [281] "2017-09-13" "2018-10-26" "2018-10-31" "2017-11-23" "2018-03-07"
## [286] "2018-01-28" "2018-02-08" "2018-03-21" "2018-12-25" "2018-03-15"
## [291] "2018-12-23" "2017-07-11" "2018-08-31" "2017-10-08" "2018-10-12"
## [296] "2018-02-14" "2018-02-13" "2017-09-04" "2018-02-15" "2018-05-02"
## [301] "2018-05-03" "2018-03-17" "2018-02-25" "2017-12-01" "2018-09-21"
## [306] "2018-09-01" "2018-01-03" "2018-04-12" "2018-12-27" "2018-06-04"
## [311] "2018-05-25" "2018-12-24" "2018-09-28" "2017-11-06" "2017-12-06"
## [316] "2018-07-19" "2018-11-26" "2017-12-11" "2017-10-01" "2018-05-26"
## [321] "2018-01-04" "2018-09-12" "2017-11-03" "2018-02-21" "2018-05-24"
## [326] "2017-08-20" "2018-06-23" "2017-10-28" "2018-05-08" "2018-02-12"
## [331] "2018-02-07" "2018-07-18" "2017-07-18" "2018-06-02" "2018-01-08"
## [336] "2017-09-20" "2017-09-24" "2018-01-15" "2017-09-14" "2017-10-18"
## [341] "2018-05-10" "2017-08-27" "2018-01-14" "2018-09-23" "2018-08-20"
## [346] "2017-10-29" "2017-08-05" "2018-10-29" "2018-08-04" "2017-08-03"
## [351] "2018-07-03" "2017-12-09" "2018-12-06" "2018-02-01" "2018-08-02"
## [356] "2018-12-12" "2018-07-12" "2017-09-01" "2017-12-10" "2018-03-02"
## [361] "2018-04-19" "2017-07-05" "2018-02-17" "2018-05-07" "2018-01-25"
## [366] "2018-07-15" "2017-10-12" "2017-09-29" "2018-01-24" "2018-03-27"
## [371] "2017-08-17" "2017-08-21" "2017-12-31" "2018-11-09" "2017-07-06"
## [376] "2018-11-08" "2018-07-22" "2018-05-18" "2018-08-27" "2018-10-09"
## [381] "2018-03-06" "2018-09-17" "2018-01-19" "2018-09-25" "2018-05-06"
## [386] "2018-07-07" "2018-06-22" "2018-02-06" "2018-05-17" "2018-09-22"
## [391] "2018-08-14" "2017-09-03" "2017-12-04" "2017-07-25" "2017-11-05"
## [396] "2018-06-09" "2018-11-07" "2018-04-18" "2018-11-23" "2018-01-27"
## [401] "2017-11-28" "2017-08-28" "2018-02-11" "2017-12-18" "2018-03-26"
## [406] "2018-10-24" "2018-01-07" "2018-11-22" "2017-08-22" "2017-09-28"
## [411] "2017-08-29" "2018-08-26" "2017-08-06" "2018-04-16" "2018-02-09"
## [416] "2018-12-14" "2018-09-20" "2018-10-30" "2018-07-24" "2018-11-20"
## [421] "2018-07-04" "2017-11-30" "2018-03-22" "2018-08-07" "2018-12-04"
## [426] "2018-03-28" "2018-05-15" "2018-12-20" "2017-08-01" "2017-11-01"
## [431] "2017-12-19" "2018-05-23" "2017-11-04" "2017-07-07" "2017-07-23"
## [436] "2017-11-13" "2018-07-20" "2018-12-11" "2018-12-17" "2018-01-31"
## [441] "2018-02-23" "2017-08-19" "2018-12-01" "2017-11-10" "2018-08-16"
## [446] "2018-02-10" "2018-05-28" "2017-11-19" "2018-09-10" "2018-09-24"
## [451] "2018-01-22" "2018-07-31" "2018-01-05" "2017-08-09" "2018-01-13"
## [456] "2017-08-30" "2018-11-28" "2017-12-23" "2017-11-25" "2017-12-16"
## [461] "2018-01-21" "2018-11-21" "2018-01-26" "2018-12-19" "2018-12-15"
## [466] "2017-11-12" "2018-08-05" "2018-08-23" "2018-03-12" "2018-07-29"
## [471] "2017-11-08" "2017-11-27" "2018-11-30" "2017-10-11" "2018-08-11"
## [476] "2018-11-15" "2018-07-16" "2017-12-22" "2017-08-07" "2018-10-25"
## [481] "2017-12-03" "2017-11-07" "2017-10-21" "2017-08-24" "2018-03-13"
## [486] "2018-01-12" "2018-11-27" "2017-09-06" "2017-07-29" "2017-08-31"
## [491] "2018-10-08" "2017-08-15" "2017-07-13" "2017-07-15" "2017-12-28"
## [496] "2017-09-23" "2017-12-20" "2017-10-26" "2018-11-29" "2017-07-09"
## [501] "2018-02-18" "2017-10-03" "2017-11-24" "2018-01-30" "2018-11-24"
## [506] "2017-08-13" "2018-01-01" "2017-12-25" "2017-11-21" "2017-12-02"
## [511] "2017-12-12" "2017-08-04" "2017-09-12" "2017-09-27" "2017-11-22"
## [516] "2018-01-17" "2017-12-13" "2017-12-08" "2017-07-31" "2018-01-18"
## [521] "2018-01-23" "2017-11-16" "2017-11-17" "2017-12-14" "2017-09-26"
## [526] "2018-01-10" "2017-07-02" "2017-07-04" "2017-07-26" "2018-01-11"
## [531] "2017-08-16" "2017-12-21" "2018-02-02" "2017-10-27" "2017-07-10"
## [536] "2017-07-30" "2017-07-08" "2017-11-26" "2017-07-20" "2017-07-22"
## [541] "2017-07-12" "2017-11-29" "2017-07-28" "2017-07-19" "2017-07-14"
## [546] "2017-08-02" "2017-07-03"
```

```
unique(training_set$market_segment_type)
```

```
## [1] "offline"       "online"        "corporate"     "aviation"
## [5] "complementary"
```

```
top_20_dates <- training_set %>%
  group_by(arrival_date) %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  select(arrival_date) %>%
  top_n(20)
```

```
## Selecting by arrival_date
```

```
training_set$arrival_date <- ifelse(training_set$arrival_date %in% top_20_dates$arrival_date,
                                    training_set$arrival_date,
                                    "other")

training_set$type_of_meal_plan <- factor(training_set$type_of_meal_plan)
training_set$room_type_reserved <- factor(training_set$room_type_reserved)
training_set$arrival_date <- factor(training_set$arrival_date)
training_set$market_segment_type <- factor(training_set$market_segment_type)

class(training_set$type_of_meal_plan)
```

```
## [1] "factor"
```

```
class(training_set$room_type_reserved)
```

```
## [1] "factor"
```

```
class(training_set$arrival_date)
```

```
## [1] "factor"
```

```
class(training_set$market_segment_type)
```

```
## [1] "factor"
```

```
levels(training_set$type_of_meal_plan)
```

```
## [1] "meal_plan_1"  "meal_plan_2"  "meal_plan_3"  "not_selected"
```

```
levels(training_set$room_type_reserved)
```

```
## [1] "room_type1" "room_type2" "room_type3" "room_type4" "room_type5"
## [6] "room_type6" "room_type7"
```

```
levels(training_set$arrival_date)
```

```
##  [1] "17877" "17878" "17879" "17880" "17881" "17882" "17883" "17884" "17885"
## [10] "17886" "17887" "17888" "17889" "17890" "17891" "17892" "17893" "17894"
## [19] "17895" "17896" "other"
```

```
levels(training_set$market_segment_type)
```

```
## [1] "aviation"      "complementary" "corporate"     "offline"
## [5] "online"
```

```
#using one hot encoding to create numerical values from non-numerical variables
onehot_encoder <- dummyVars(~ type_of_meal_plan + room_type_reserved +
                              arrival_date + market_segment_type,
                            training_set[, c("type_of_meal_plan",
                                             "room_type_reserved",
                                             "arrival_date",
                                             "market_segment_type")],
                            levelsOnly = TRUE,
                            fullRank = TRUE)
```

```
onehot_enc_training <- predict(onehot_encoder,
                               training_set[, c("type_of_meal_plan",
                                                "room_type_reserved",
                                                "arrival_date",
                                                "market_segment_type")])

training_set <- cbind(training_set, onehot_enc_training)

test_set$arrival_date <- ifelse(test_set$arrival_date %in%
                                  top_20_dates$arrival_date,
                                test_set$arrival_date,
                                "other")

test_set$type_of_meal_plan <- factor(test_set$type_of_meal_plan)
test_set$room_type_reserved <- factor(test_set$room_type_reserved)
test_set$arrival_date <- factor(test_set$arrival_date)
test_set$market_segment_type <- factor(test_set$market_segment_type)

onehot_enc_test <- predict(onehot_encoder, test_set[, c("type_of_meal_plan",
                                                        "room_type_reserved",
                                                        "arrival_date",
                                                        "market_segment_type")])
test_set <- cbind(test_set, onehot_enc_test)

#scaling and centering variables to create consistent results
test_set[,-c(1, 6, 8, 10, 11, 17)] <- scale(test_set[,-c(1, 6, 8, 10, 11, 17)],
                                            center = apply(training_set[,-c(1, 6, 8, 10, 11, 17)], 2, me
                                            scale = apply(training_set[,-c(1, 6, 8, 10, 11, 17)], 2, sd)
training_set[,-c(1, 6, 8, 10, 11, 17)] <- scale(training_set[,-c(1, 6, 8, 10, 11, 17)])

training_features <- array(data = unlist(training_set[,-c(1, 6, 8, 10, 11, 17)]),
                           dim = c(nrow(training_set), 44))
training_labels <- array(data = unlist(training_set[, 17]),
                         dim = c(nrow(training_set)))
test_features <- array(data = unlist(test_set[,-c(1, 6, 8, 10, 11, 17)]),
                       dim = c(nrow(test_set), 44))
test_labels <- array(data = unlist(test_set[, 17]),
                     dim = c(nrow(test_set)))
```