5

Ma Lok Sum, Zoe (a1819866)

2022-10-21

```
# Get the data
library(readr)
affairs <- read_csv("Downloads/affairs.csv")
```

```
## Rows: 570 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): sex, child
## dbl (7): affair, age, ym, religious, education, occupation, rate
##
## 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.
```

```
affairs
```

```
## # A tibble: 570 x 9
##    affair sex      age    ym child religious education occupation  rate
##     <dbl> <chr>  <dbl> <dbl> <chr>     <dbl>     <dbl>      <dbl> <dbl>
##  1      0 female    32   7   yes           4        17          5     4
##  2      0 male      27   1.5 yes           2        17          4     4
##  3      0 female    22   1.5 no            3        16          5     3
##  4      1 female    27   4   yes           3        17          1     5
##  5      1 female    27   4   no            2        14          5     5
##  6      0 male      37  15   yes           4        17          5     3
##  7      0 male      27   4   yes           3        20          6     5
##  8      0 female    27   4   yes           2        16          1     4
##  9      0 male      57  15   yes           5        18          5     2
## 10      0 female    52  15   yes           3        16          5     4
## # ... with 560 more rows
```

Data cleaning

1.

```
# read the data as a tibble
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v dplyr   1.0.10
## v tibble  3.1.8      v stringr 1.4.1
## v tidyr   1.2.1      v forcats 0.5.2
## v purrr   0.3.5
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
as_tibble(affairs)
```

```
## # A tibble: 570 x 9
##    affair sex      age    ym child religious education occupation  rate
##     <dbl> <chr>  <dbl> <dbl> <chr>     <dbl>     <dbl>      <dbl> <dbl>
##  1      0 female    32   7   yes           4        17          5     4
##  2      0 male      27   1.5 yes           2        17          4     4
##  3      0 female    22   1.5 no            3        16          5     3
##  4      1 female    27   4   yes           3        17          1     5
##  5      1 female    27   4   no            2        14          5     5
##  6      0 male      37  15   yes           4        17          5     3
##  7      0 male      27   4   yes           3        20          6     5
##  8      0 female    27   4   yes           2        16          1     4
##  9      0 male      57  15   yes           5        18          5     2
## 10      0 female    52  15   yes           3        16          5     4
## # ... with 560 more rows
```

```
# display the first 6 rows
head(affairs)
```

```
## # A tibble: 6 x 9
##   affair sex      age    ym child religious education occupation  rate
##    <dbl> <chr>  <dbl> <dbl> <chr>     <dbl>     <dbl>      <dbl> <dbl>
## 1      0 female    32   7   yes           4        17          5     4
## 2      0 male      27   1.5 yes           2        17          4     4
## 3      0 female    22   1.5 no            3        16          5     3
## 4      1 female    27   4   yes           3        17          1     5
## 5      1 female    27   4   no            2        14          5     5
## 6      0 male      37  15   yes           4        17          5     3
```

2.

The outcome variable is affair.

The predictor variables are sex, age, ym, child,religious, education, occupation and rate.

3.

```
# skim the data
library(skimr)
skim(affairs)
```

Table 1: Data summary

Name	affairs
Number of rows	570
Number of columns	9
Column type frequency:	
character	2
numeric	7

Table 1: Data summary

Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
sex	0	1	4	6	0	2	0
child	0	1	2	3	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
affair	0	1	0.25	0.44	0.00	0	0	1	1	
age	0	1	32.30	9.23	17.50	27	32	37	57	
ym	0	1	8.06	5.55	0.12	4	7	15	15	
religious	0	1	3.12	1.17	1.00	2	3	4	5	
education	0	1	16.18	2.38	9.00	14	16	18	20	
occupation	0	1	4.18	1.82	1.00	3	5	6	7	
rate	0	1	3.94	1.09	1.00	3	4	5	5	

There is no missing data.

There are 570 observations on 9 variables.

There is one variable (affair) been read in incorrectly.

4.

```r
# Convert the affair variable to a yes/ no response
affairs$affair [affairs$affair == 1] <- "Yes"
affairs$affair [affairs$affair == 0] <- "No"

# Change all character variables to factors
affairs$sex <- as.factor(affairs$sex)
affairs$child <- as.factor(affairs$child)
affairs$affair <- as.factor(affairs$affair)
affairs
```

```
## # A tibble: 570 x 9
##    affair sex      age    ym child religious education occupation  rate
##    <fct>  <fct>  <dbl> <dbl> <fct>     <dbl>     <dbl>      <dbl> <dbl>
##  1 No     female    32   7   yes           4        17          5     4
##  2 No     male      27   1.5 yes           2        17          4     4
##  3 No     female    22   1.5 no            3        16          5     3
##  4 Yes    female    27   4   yes           3        17          1     5
##  5 Yes    female    27   4   no            2        14          5     5
##  6 No     male      37  15   yes           4        17          5     3
##  7 No     male      27   4   yes           3        20          6     5
##  8 No     female    27   4   yes           2        16          1     4
##  9 No     male      57  15    yes             5         18            5     2
## 10 No     female    52  15    yes             3         16            5     4
## # ... with 560 more rows
```

5.

```
# skim the data
skim(affairs)
```

Table 4: Data summary

Name	affairs
Number of rows	570
Number of columns	9
Column type frequency:	
factor	3
numeric	6
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
affair	0	1	FALSE	2	No: 425, Yes: 145
sex	0	1	FALSE	2	fem: 299, mal: 271
child	0	1	FALSE	2	yes: 406, no: 164

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
age	0	1	32.30	9.23	17.50	27	32	37	57	
ym	0	1	8.06	5.55	0.12	4	7	15	15	
religious	0	1	3.12	1.17	1.00	2	3	4	5	
education	0	1	16.18	2.38	9.00	14	16	18	20	
occupation	0	1	4.18	1.82	1.00	3	5	6	7	
rate	0	1	3.94	1.09	1.00	3	4	5	5	

a. There are 145 people responded as having had an affair.

There are 406 people responded to having children.

b.The mean age of respondents are 32.3.

The mean response on the religious scale is 3.12.

Exploratory analysis

1

```
# count the group first
library(plyr)
```

```
## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following object is masked from 'package:purrr':
##
##     compact
```

```
library(dplyr)
affairs %>%
  group_by(affair, sex)%>%
  dplyr::summarize(count=n())
```

```
## `summarise()` has grouped output by 'affair'. You can override using the
## `.groups` argument.

## # A tibble: 4 x 3
## # Groups:   affair [2]
##   affair sex    count
##   <fct>  <fct>  <int>
## 1 No     female   228
## 2 No     male     197
## 3 Yes    female    71
## 4 Yes    male      74
```

```
228/(228+197)
```

```
## [1] 0.5364706
```

The proportion of the female participants haven't have an affair is 0.536.

```
71/(71+74)
```

```
## [1] 0.4896552
```

The proportion of the female participants have an affair is 0.490.

It will be a difference in the proportion of females who will have an affair as opposed to those who will not.

2

```
# count the group first
affairs %>%
  group_by(affair, child)%>%
  dplyr::summarize(count=n())
```

```
## `summarise()` has grouped output by 'affair'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 4 x 3
## # Groups:   affair [2]
##   affair child count
##   <fct>  <fct> <int>
## 1 No     no      137
## 2 No     yes     288
## 3 Yes    no       27
## 4 Yes    yes     118
```

```
118/(118+288)
```

```
## [1] 0.2906404
```

The proportion of the participants have an affair and also had children is 0.291.

```
228/(118+288)
```

```
## [1] 0.5615764
```

The proportion of the participants haven't have an affair and also had children is 0.562.

Based on the results, It is more likely to haven't children if I have an affair.

Split and preprocess

1

```
library(tidymodels)
```

```
## -- Attaching packages -------------------------------------- tidymodels 1.0.0 --
```

```
## v broom        1.0.1     v rsample      1.1.0
## v dials        1.0.0     v tune         1.0.1
## v infer         1.0.3     v workflows    1.1.0
## v modeldata     1.0.1     v workflowsets 1.0.0
## v parsnip       1.0.2     v yardstick    1.1.0
## v recipes       1.0.2
```

```
## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x plyr::arrange()   masks dplyr::arrange()
## x plyr::compact()   masks purrr::compact()
## x plyr::count()     masks dplyr::count()
## x scales::discard() masks purrr::discard()
## x plyr::failwith()  masks dplyr::failwith()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x plyr::id()        masks dplyr::id()
## x dplyr::lag()      masks stats::lag()
## x plyr::mutate()    masks dplyr::mutate()
## x plyr::rename()    masks dplyr::rename()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## x plyr::summarise() masks dplyr::summarise()
## x plyr::summarize() masks dplyr::summarize()
## * Use suppressPackageStartupMessages() to eliminate package startup messages
```

```r
# set a seed for reproducibility
set.seed(123)
# split the data
rsplit<-initial_split(affairs)
rsplit
```

```
## <Training/Testing/Total>
## <427/143/570>
```

There are 427 observations in the training set.

There are 143 observations in the testing set.

2

```r
library(caTools)
# obtain the training sets
train_ind <- training(rsplit)
# Display the first 6 rows of the training set
head(train_ind)
```

```
## # A tibble: 6 x 9
##   affair sex      age    ym child religious education occupation  rate
##   <fct>  <fct>  <dbl> <dbl> <fct>     <dbl>     <dbl>      <dbl> <dbl>
## 1 No     female  27     4    yes           2        18          6     1
## 2 No     male    37    15    yes           5        20          6     4
## 3 No     female  17.5  0.75 no            2        18          5     4
## 4 No     male    22     0.75 no            4        16          6     4
## 5 No     male    57    15    no            4         9          3     1
## 6 No     female  52    15    yes           4        14          1     3
```

```
# obtain the testing sets
test_cl <- testing(rsplit)
# display the first 6 rows of the testing sets
head(test_cl)
```

```
## # A tibble: 6 x 9
##   affair sex      age    ym child religious education occupation  rate
##   <fct>  <fct>  <dbl> <dbl> <fct>     <dbl>     <dbl>      <dbl> <dbl>
## 1 No     female    32  7    yes           4        17          5     4
## 2 No     male      27  1.5  yes           2        17          4     4
## 3 No     male      57 15    yes           5        18          5     2
## 4 No     male      27  0.75 no            2        17          5     5
## 5 Yes    male      32 10    yes           2        17          6     5
## 6 Yes    female    32 15    yes           3        14          1     5
```

3

```
library(themis)
themis::step_downsample(affairs)
```

```
## Warning: Unknown or uninitialised column: `steps`.
## Unknown or uninitialised column: `steps`.
```

```
## # A tibble: 570 x 10
##    affair sex      age    ym child religious education occupa~1  rate steps
##    <fct>  <fct>  <dbl> <dbl> <fct>     <dbl>     <dbl>    <dbl> <dbl> <list>
##  1 No     female    32   7   yes           4        17        5     4 <stp_dwns>
##  2 No     male      27   1.5 yes           2        17        4     4 <stp_dwns>
##  3 No     female    22   1.5 no            3        16        5     3 <stp_dwns>
##  4 Yes    female    27   4   yes           3        17        1     5 <stp_dwns>
##  5 Yes    female    27   4   no            2        14        5     5 <stp_dwns>
##  6 No     male      37  15   yes           4        17        5     3 <stp_dwns>
##  7 No     male      27   4   yes           3        20        6     5 <stp_dwns>
##  8 No     female    27   4   yes           2        16        1     4 <stp_dwns>
##  9 No     male      57  15   yes           5        18        5     2 <stp_dwns>
## 10 No     female    52  15   yes           3        16        5     4 <stp_dwns>
## # ... with 560 more rows, and abbreviated variable name 1: occupation
```

```
# We want to down sample because the data set is imbalance.
```

4

```
# crete a recipe
af_recipe <- recipe( affair ~ ., data = train_ind) %>%
  themis::step_downsample(affair) %>%
  step_dummy( sex, child ) %>%
  # Convert all our categorical predictors to a dummy variable
  step_normalize( all_predictors() ) # Normalize our predictors

af_recipe
```

```
## Recipe
##
```

```
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor          8
##
## Operations:
##
## Down-sampling based on affair
## Dummy variables from sex, child
## Centering and scaling for all_predictors()
```

```
# print out the recipe
af_prepped <- af_recipe %>%
  prep()
af_prepped
```

```
## Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor          8
##
## Training data contained 427 data points and no missing data.
##
## Operations:
##
## Down-sampling based on affair [trained]
## Dummy variables from sex, child [trained]
## Centering and scaling for age, ym, religious, education, occupation, rate... [trained]
```

5

a: to prepare the training dataset

```
ad_juiced <- juice( af_prepped)
ad_juiced %>%
  head()
```

```
## # A tibble: 6 x 9
##      age     ym religious education occupation  rate affair sex_male child_yes
##    <dbl>  <dbl>     <dbl>     <dbl>      <dbl> <dbl> <fct>     <dbl>     <dbl>
## 1 -1.13  -1.27    -0.803     -0.920      0.424 1.06  No        -0.927     0.608
## 2  2.62   1.19    -0.803     -0.920     -0.130 0.185 No         1.07      0.608
## 3  0.478  1.19     0.930      1.59       0.424 0.185 No         1.07      0.608
## 4 -0.592 -0.811    0.0636     0.336      0.978 1.06  No         1.07      0.608
## 5 -0.592 -0.811    0.930     -0.920      0.424 0.185 No         1.07      0.608
## 6 -1.13  -1.27     0.930     -0.920      0.424 0.185 No        -0.927    -1.64
```

b: to prepare the testing dataset

```
ad_prepped <-  af_recipe %>%
  prep()
ad_prepped
```

```
## Recipe
##
## Inputs:
##
##       role #variables
##    outcome          1
##  predictor          8
##
## Training data contained 427 data points and no missing data.
##
## Operations:
##
## Down-sampling based on affair [trained]
## Dummy variables from sex, child [trained]
## Centering and scaling for age, ym, religious, education, occupation, rate... [trained]
```

```
ad_baked <- bake( ad_prepped, test_cl )
ad_baked %>%
  head()
```

```
## # A tibble: 6 x 9
##       age     ym religious education occupation   rate affair sex_male child_yes
##     <dbl>  <dbl>     <dbl>     <dbl>      <dbl>  <dbl> <fct>     <dbl>     <dbl>
## 1 -0.0574 -0.264    0.930      0.336      0.424  0.185 No       -0.927     0.608
## 2 -0.592  -1.27    -0.803      0.336     -0.130  0.185 No        1.07      0.608
## 3  2.62    1.19     1.80       0.755      0.424 -1.57  No        1.07      0.608
## 4 -0.592  -1.40    -0.803      0.336      0.424  1.06  No        1.07     -1.64
## 5 -0.0574  0.282   -0.803      0.336      0.978  1.06  Yes       1.07      0.608
## 6 -0.0574  1.19     0.0636    -0.920     -1.79   1.06  Yes      -0.927     0.608
```

6

```
skim(ad_juiced)
```

Table 7: Data summary

Name	ad_juiced
Number of rows	218
Number of columns	9
Column type frequency:	
factor	1
numeric	8
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
affair	0	1	FALSE	2	No: 109, Yes: 109

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
age	0	1	0	1	-1.61	-0.59	-0.06	0.48	2.62	
ym	0	1	0	1	-1.52	-0.81	-0.26	1.19	1.19	
religious	0	1	0	1	-1.67	-0.80	0.06	0.93	1.80	
education	0	1	0	1	-3.02	-0.92	0.34	0.76	1.59	
occupation	0	1	0	1	-1.79	-0.68	0.42	0.98	1.53	
rate	0	1	0	1	-2.45	-0.69	0.19	1.06	1.06	
sex_male	0	1	0	1	-0.93	-0.93	-0.93	1.07	1.07	
child_yes	0	1	0	1	-1.64	-1.64	0.61	0.61	0.61	

```
# It has done what I expect because the mean is close to standard deviation
```

Tune and fit a model

1

```
library(class)
# Fitting KNN Models to training dataset
kknn_spec <- nearest_neighbor( mode = "classification", neighbors = tune() ) %>%
  set_engine( "kknn" )
kknn_spec
```

```
## K-Nearest Neighbor Model Specification (classification)
##
## Main Arguments:
##   neighbors = tune()
##
## Computational engine: kknn
```

2

```
# Create a 5-fold cross validation set
set.seed(1223)
train_cv <- vfold_cv( ad_juiced, v = 5, strata = affair)
train_cv
```

```
## #  5-fold cross-validation using stratification
## # A tibble: 5 x 2
##   splits           id
##   <list>           <chr>
## 1 <split [174/44]> Fold1
## 2 <split [174/44]> Fold2
## 3 <split [174/44]> Fold3
## 4 <split [174/44]> Fold4
## 5 <split [176/42]> Fold5
```

3

```
# make a grid of k-values to tune our model on using levels 25 and range from 5 to 75
affair_grid <- grid_regular( neighbors (range(5:75)),
                             levels = 25)
affair_grid
```

```
## # A tibble: 25 x 1
##    neighbors
##        <int>
##  1         5
##  2         7
##  3        10
##  4        13
##  5        16
##  6        19
##  7        22
##  8        25
##  9        28
## 10        31
## # ... with 15 more rows
```

4

```
# tune k-nearest neighbours model using the cross validation sets and grid of k-values.
library(kknn)
knn_tuned <- tune_grid( object = kknn_spec,
                       preprocessor = recipe(affair ~ .,ad_juiced),
                       resamples = train_cv,
                       grid = affair_grid)
knn_tuned
```

```
## # Tuning results
## # 5-fold cross-validation using stratification
## # A tibble: 5 x 4
##   splits            id    .metrics          .notes
##   <list>            <chr> <list>            <list>
## 1 <split [174/44]> Fold1 <tibble [50 x 5]> <tibble [0 x 3]>
## 2 <split [174/44]> Fold2 <tibble [50 x 5]> <tibble [0 x 3]>
## 3 <split [174/44]> Fold3 <tibble [50 x 5]> <tibble [0 x 3]>
## 4 <split [174/44]> Fold4 <tibble [50 x 5]> <tibble [0 x 3]>
## 5 <split [176/42]> Fold5 <tibble [50 x 5]> <tibble [0 x 3]>
```

5

```
best_auc <- select_best( knn_tuned, "accuracy" )
best_auc
```

```
## # A tibble: 1 x 2
##   neighbors .config
##       <int> <chr>
## 1        10 Preprocessor1_Model03
```

```
#  the best k values(neignbour) is 10
```

6

```
# apply the best k values to the model
final_knn <- finalize_model(kknn_spec, best_auc)
final_knn
```

```
## K-Nearest Neighbor Model Specification (classification)
##
## Main Arguments:
##   neighbors = 10
##
## Computational engine: kknn
```

7

```
# the final model
affairs_knn <- final_knn %>%
  fit( affair ~ . , data = ad_juiced )
affairs_knn
```

```
## parsnip model object
##
##
## Call:
## kknn::train.kknn(formula = affair ~ ., data = data, ks = min_rows(10L,     data, 5))
##
## Type of response variable: nominal
## Minimal misclassification: 0.3807339
## Best kernel: optimal
## Best k: 10
```

Evaluation

1

```
# Obtain class predictions
knn_preds <- predict( affairs_knn,
                      new_data = ad_baked,
                      type = "class" ) %>%
  bind_cols(ad_baked %>%
             dplyr::select(affair) )

knn_preds %>%
  metrics( truth = affair, estimate = .pred_class )
```

```
## # A tibble: 2 x 3
##   .metric  .estimator .estimate
##   <chr>    <chr>          <dbl>
## 1 accuracy binary         0.545
## 2 kap      binary        0.0194
```

2

```r
# add the true value of affair
knn_preds %>%
  conf_mat(truth = affair, estimate = .pred_class)
```

```
##             Truth
## Prediction No Yes
##        No  62  20
##        Yes 45  16
```

```r
head(knn_preds)
```

```
## # A tibble: 6 x 2
##   .pred_class affair
##   <fct>       <fct>
## 1 Yes         No
## 2 Yes         No
## 3 No          No
## 4 No          No
## 5 No          Yes
## 6 No          Yes
```

3

```r
# Confusion Matrix
cm <- knn_preds %>% conf_mat(truth = affair, estimate = .pred_class)
cm
```

```
##             Truth
## Prediction No Yes
##        No  62  20
##        Yes 45  16
```

4

```r
sens <- 62/(62 + 45)
spec <- 20/(20 + 16)

tibble( sensitivity = sens,
        specificity = spec)
```

```
## # A tibble: 1 x 2
##   sensitivity specificity
##         <dbl>       <dbl>
## 1       0.579       0.556
```

It is better if we can adjusted since the sensitivity and specificity is not really good to tell us the cut off point.

5

a

```
# input the new information--Kelvin
new_tibble <-
 tibble(sex = "male", age = 45, ym = 15, child = "yes", religious = 1,
        education = 20, occupation = 7, rate = 3 )
new_tibble
```

```
## # A tibble: 1 x 8
##   sex     age    ym child religious education occupation  rate
##   <chr> <dbl> <dbl> <chr>     <dbl>     <dbl>      <dbl> <dbl>
## 1 male     45    15 yes           1        20          7     3
```

b

```
# preprocess Kevin's information with my recipe
ad_baked <- bake( ad_prepped, new_tibble )
```

```
## Warning:  There were 2 columns that were factors when the recipe was prepped:
##  'sex', 'child'.
##  This may cause errors when processing new data.
```

```
ad_baked %>%
  head()
```

```
## # A tibble: 1 x 8
##     age    ym religious education occupation   rate sex_male child_yes
##   <dbl> <dbl>     <dbl>     <dbl>      <dbl>  <dbl>    <dbl>     <dbl>
## 1  1.33  1.19     -1.67      1.59       1.53 -0.692     1.07     0.608
```

```
ad_baked
```

```
## # A tibble: 1 x 8
##     age    ym religious education occupation   rate sex_male child_yes
##   <dbl> <dbl>     <dbl>     <dbl>      <dbl>  <dbl>    <dbl>     <dbl>
## 1  1.33  1.19     -1.67      1.59       1.53 -0.692     1.07     0.608
```

c

```
# obtain a predicted probability
predict(affairs_knn, new_data = ad_baked, type = "prob")
```

```
## # A tibble: 1 x 2
##   .pred_No .pred_Yes
##      <dbl>     <dbl>
## 1    0.301     0.699
```

d

Based on the results above, I am comfortable going to Kevin's partner with my prediction of Kevin will have an affair (0.699 predicted probabilities).