Data Science Template
End-to-End **weatherAUS** Analysis

Graham Williams

10th September 2018

This template provides a minimal example of a data science template. We will build a decision tree model to predict future events from historic observations of that event. In this case the event is whether it will rain tomorrow.

The concept of templates for Data Science are developed in the book The Essentials of Data Science (2017). The actual source files and scripts, with regular updates, available from the Essentials web site (essentials.togaware.com).

As with all of our templates and reports we collect up front here the packages used to support the creation of this document.

```
# Load required packages from local library into R.

library(magrittr)     # Pipe operator %>% %<>% %T>% equals().
library(lubridate)    # Dates and time.
library(rattle)       # normVarNames().
library(ROCR)         # Use prediction() for evaluation.
library(rpart)        # Model: decision tree.
library(scales)       # Include commas in numbers.
library(stringi)      # String concat operator %s+%.
library(tidyverse)    # ggplot2, tibble, tidyr, readr, purr, dplyr, stringr
```

1 Data Source

```
# Original dataset source/location.

dsorig <- file.path("https://rattle.togaware.com/weatherAUS.csv")

# Name of the dataset.

dsname <- "weatherAUS"

# Identify the Essentials location of the dataset.

dsloc  <- "https://essentials.togaware.com"
dspath <- file.path(dsloc, dsname %s+% ".csv") %T>% print()
## [1] "https://essentials.togaware.com/weatherAUS.csv"
```

2 Data Ingestion

```
# Ingest the dataset.

dspath %>% read_csv() %>% assign(dsname, ., envir=.GlobalEnv)
```

3 Generic Template Variable and Initial View

```
# Store the dataset with a generic template variable name.

dsname %>% get() %T>% print() -> ds
```

```
## # A tibble: 145,463 x 24
##    Date       Location MinTemp MaxTemp Rainfall Evaporation Sunshine
##    <date>     <chr>      <dbl>   <dbl>    <dbl> <chr>       <chr>
##  1 2008-12-01 Albury      13.4    22.9      0.6 <NA>        <NA>
##  2 2008-12-02 Albury       7.4    25.1      0   <NA>        <NA>
##  3 2008-12-03 Albury      12.9    25.7      0   <NA>        <NA>
##  4 2008-12-04 Albury       9.2    28        0   <NA>        <NA>
##  5 2008-12-05 Albury      17.5    32.3      1   <NA>        <NA>
##  6 2008-12-06 Albury      14.6    29.7      0.2 <NA>        <NA>
##  7 2008-12-07 Albury      14.3    25        0   <NA>        <NA>
##  8 2008-12-08 Albury       7.7    26.7      0   <NA>        <NA>
##  9 2008-12-09 Albury       9.7    31.9      0   <NA>        <NA>
## 10 2008-12-10 Albury      13.1    30.1      1.4 <NA>        <NA>
## # ... with 145,453 more rows, and 17 more variables: WindGustDir <chr>,
## #   WindGustSpeed <int>, WindDir9am <chr>, WindDir3pm <chr>,
## #   WindSpeed9am <int>, WindSpeed3pm <int>, Humidity9am <int>,
## #   Humidity3pm <int>, Pressure9am <dbl>, Pressure3pm <dbl>,
## #   Cloud9am <int>, Cloud3pm <int>, Temp9am <dbl>, Temp3pm <dbl>,
## #   RainToday <chr>, RISK_MM <dbl>, RainTomorrow <chr>
```

4 Normalise Variable Names

```
# Normalise the variable names.

names(ds) %<>% normVarNames() %T>% print()
```

```
##  [1] "date"            "location"        "min_temp"
##  [4] "max_temp"        "rainfall"        "evaporation"
##  [7] "sunshine"        "wind_gust_dir"   "wind_gust_speed"
## [10] "wind_dir_9am"    "wind_dir_3pm"    "wind_speed_9am"
## [13] "wind_speed_3pm"  "humidity_9am"    "humidity_3pm"
## [16] "pressure_9am"    "pressure_3pm"    "cloud_9am"
## [19] "cloud_3pm"       "temp_9am"        "temp_3pm"
## [22] "rain_today"      "risk_mm"         "rain_tomorrow"
```

```
# Fix specific variable names.

names(ds)[23] <- c("rainfall_tomorrow")

# Check the names.

names(ds)
```

```
##  [1] "date"              "location"          "min_temp"
##  [4] "max_temp"          "rainfall"          "evaporation"
##  [7] "sunshine"          "wind_gust_dir"     "wind_gust_speed"
## [10] "wind_dir_9am"      "wind_dir_3pm"      "wind_speed_9am"
## [13] "wind_speed_3pm"    "humidity_9am"      "humidity_3pm"
## [16] "pressure_9am"      "pressure_3pm"      "cloud_9am"
## [19] "cloud_3pm"         "temp_9am"          "temp_3pm"
## [22] "rain_today"        "rainfall_tomorrow" "rain_tomorrow"
```

5 Key Variables

```
# Note any identifiers.

id <- c("date", "location")

# Note the target variable.

target <- "rain_tomorrow"

# Note any risk variable - measures the severity of the outcome.

risk <- "rainfall_tomorrow"
```

6 Variables for Analysis

```
# Note available variables ignoring identifiers and risk, with target first.

ds %>%
  names() %>%
  setdiff(c(id, risk)) %>%
  c(target, .) %>%
  unique() %T>%
  print() ->
vars
```

```
##  [1] "rain_tomorrow"   "min_temp"        "max_temp"
##  [4] "rainfall"        "evaporation"     "sunshine"
##  [7] "wind_gust_dir"   "wind_gust_speed" "wind_dir_9am"
## [10] "wind_dir_3pm"    "wind_speed_9am"  "wind_speed_3pm"
## [13] "humidity_9am"    "humidity_3pm"    "pressure_9am"
## [16] "pressure_3pm"    "cloud_9am"       "cloud_3pm"
## [19] "temp_9am"        "temp_3pm"        "rain_today"
```

7 Variables to Ignore

```
# We will sometimes want to ignore specific variables.

ignore <- NULL

# Remove variables to ignore from the variable list.

vars %<>% setdiff(ignore) %T>% print()
```

```
##  [1] "rain_tomorrow"   "min_temp"        "max_temp"
##  [4] "rainfall"        "evaporation"     "sunshine"
##  [7] "wind_gust_dir"   "wind_gust_speed" "wind_dir_9am"
## [10] "wind_dir_3pm"    "wind_speed_9am"  "wind_speed_3pm"
## [13] "humidity_9am"    "humidity_3pm"    "pressure_9am"
## [16] "pressure_3pm"    "cloud_9am"       "cloud_3pm"
## [19] "temp_9am"        "temp_3pm"        "rain_today"
```

8 Deal with Numeric Variables

In our dataset we might notice that evaporation and sunshine have come through as character variables. This is because there are only missing values in the first many observations and the heuristics interprets that as character variables.

```
# Fix specific numeric variables.

cvars <- c("evaporation", "sunshine")
ds[cvars] %<>% sapply(as.numeric)

# Remove observations without a value for rain_tomorrow.

ds %<>% drop_na(rain_tomorrow)
```

9 Deal with Character Variables

```
# Identify the character variables by index.

chari <- ds[vars] %>% sapply(is.character) %>% which() %T>% print()
```

```
## rain_tomorrow wind_gust_dir  wind_dir_9am  wind_dir_3pm    rain_today
##             1             7             9            10            21
```

```
# Identify the character variables by name.

charc <- ds[vars] %>% names() %>% '['(chari) %T>% print()
```

```
## [1] "rain_tomorrow" "wind_gust_dir" "wind_dir_9am"  "wind_dir_3pm"
## [5] "rain_today"
```

10 Character Variable Levels

```
# Observe the unique levels.

ds[charc] %>% sapply(unique)
```

```
## $rain_tomorrow
## [1] "No"  "Yes"
##
## $wind_gust_dir
##  [1] "W"   "WNW" "WSW" "NE"  "NNW" "N"   "NNE" "SW"  "ENE" "SSE" "S"
## [12] "NW"  "SE"  "ESE" NA    "E"   "SSW"
##
## $wind_dir_9am
##  [1] "W"   "NNW" "SE"  "ENE" "SW"  "SSE" "S"   "NE"  NA    "SSW" "N"
## [12] "WSW" "ESE" "E"   "NW"  "WNW" "NNE"
##
## $wind_dir_3pm
##  [1] "WNW" "WSW" "E"   "NW"  "W"   "SSE" "ESE" "ENE" "NNW" "SSW" "SW"
## [12] "SE"  "N"   "S"   "NNE" NA    "NE"
##
## $rain_today
## [1] "No"  "Yes" NA
```

11 Characters to Factors

```
# Convert all character to factor if determined appropriate.

ds[charc] %<>% map(factor)
```

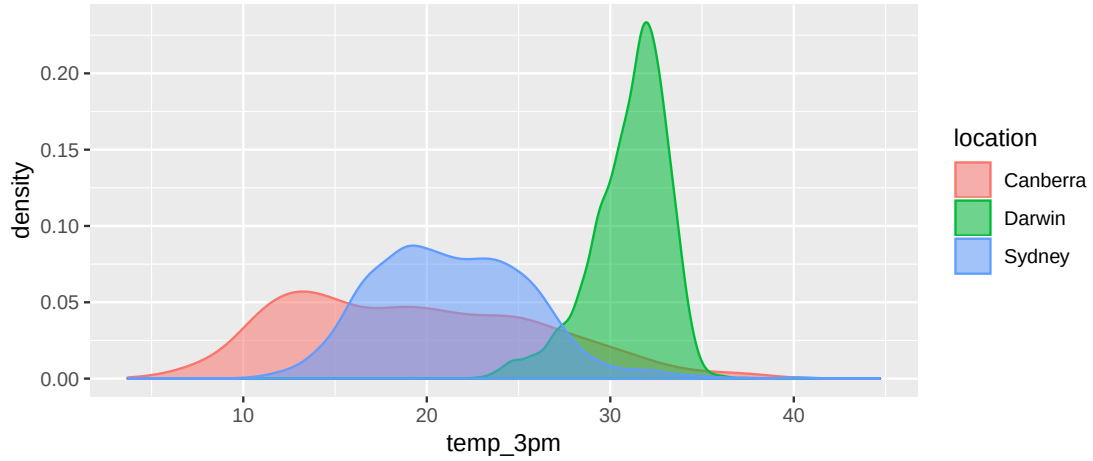
12 Data Observation

```
# A glimpse into the dataset.

glimpse(ds)
```

```
## Observations: 142,246
## Variables: 24
## $ date              <date> 2008-12-01, 2008-12-02, 2008-12-03, 2008-12...
## $ location          <chr> "Albury", "Albury", "Albury", "Albury", "Alb...
## $ min_temp          <dbl> 13.4, 7.4, 12.9, 9.2, 17.5, 14.6, 14.3, 7.7,...
## $ max_temp          <dbl> 22.9, 25.1, 25.7, 28.0, 32.3, 29.7, 25.0, 26...
## $ rainfall          <dbl> 0.6, 0.0, 0.0, 0.0, 1.0, 0.2, 0.0, 0.0, 0.0,...
## $ evaporation       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ sunshine          <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ...
## $ wind_gust_dir     <fct> W, WNW, WSW, NE, W, WNW, W, W, NNW, W, N, NN...
## $ wind_gust_speed   <int> 44, 44, 46, 24, 41, 56, 50, 35, 80, 28, 30, ...
## $ wind_dir_9am      <fct> W, NNW, W, SE, ENE, W, SW, SSE, SE, S, SSE, ...
## $ wind_dir_3pm      <fct> WNW, WSW, WSW, E, NW, W, W, W, NW, SSE, ESE,...
## $ wind_speed_9am    <int> 20, 4, 19, 11, 7, 19, 20, 6, 7, 15, 17, 15, ...
## $ wind_speed_3pm    <int> 24, 22, 26, 9, 20, 24, 24, 17, 28, 11, 6, 13...
## $ humidity_9am      <int> 71, 44, 38, 45, 82, 55, 49, 48, 42, 58, 48, ...
## $ humidity_3pm      <int> 22, 25, 30, 16, 33, 23, 19, 19, 9, 27, 22, 9...
## $ pressure_9am      <dbl> 1007.7, 1010.6, 1007.6, 1017.6, 1010.8, 1009...
## $ pressure_3pm      <dbl> 1007.1, 1007.8, 1008.7, 1012.8, 1006.0, 1005...
## $ cloud_9am         <int> 8, NA, NA, NA, 7, NA, 1, NA, NA, NA, NA, 8, ...
## $ cloud_3pm         <int> NA, NA, 2, NA, 8, NA, NA, NA, NA, NA, NA, 8,...
## $ temp_9am          <dbl> 16.9, 17.2, 21.0, 18.1, 17.8, 20.6, 18.1, 16...
## $ temp_3pm          <dbl> 21.8, 24.3, 23.2, 26.5, 29.7, 28.9, 24.6, 25...
## $ rain_today        <fct> No, No, No, No, No, No, No, No, No, Yes, No,...
## $ rainfall_tomorrow <dbl> 0.0, 0.0, 0.0, 1.0, 0.2, 0.0, 0.0, 0.0, 1.4,...
## $ rain_tomorrow     <fct> No, No, No, No, No, No, No, No, Yes, No, Yes...
```

13 Data Visualisation

```
# Visualise relationships in the data.

ds %>%
  filter(location %in% c("Canberra", "Darwin", "Sydney")) %>%
  filter(temp_3pm %>% is.na() %>% not()) %>%
  select(temp_3pm, location) %>%
  ggplot(aes(x=temp_3pm, colour=location, fill=location)) +
  geom_density(alpha=0.55)
```



14 Model Formula

```
# Formula for modelling.

ds[vars] %>%
  formula() %T>%
  print() ->
form
```

```
## rain_tomorrow ~ min_temp + max_temp + rainfall + evaporation +
##     sunshine + wind_gust_dir + wind_gust_speed + wind_dir_9am +
##     wind_dir_3pm + wind_speed_9am + wind_speed_3pm + humidity_9am +
##     humidity_3pm + pressure_9am + pressure_3pm + cloud_9am +
##     cloud_3pm + temp_9am + temp_3pm + rain_today
## <environment: 0x562c6a4063d0>
```

15 Target as a Categoric

```
# Ensure the target is categoric.

ds[[target]] %<>% factor()
```

16 Variables and Observations

```
# Identify the input variables by name.

inputs <- setdiff(vars, target) %T>% print()
```

```
##  [1] "min_temp"       "max_temp"       "rainfall"
##  [4] "evaporation"    "sunshine"       "wind_gust_dir"
##  [7] "wind_gust_speed" "wind_dir_9am"   "wind_dir_3pm"
## [10] "wind_speed_9am"  "wind_speed_3pm" "humidity_9am"
## [13] "humidity_3pm"    "pressure_9am"   "pressure_3pm"
## [16] "cloud_9am"       "cloud_3pm"      "temp_9am"
## [19] "temp_3pm"        "rain_today"
```

```
# Record the number of observations.

nobs <- nrow(ds) %T>% comcat()
```

```
## 142,246
```

17 Training Dataset

```
# Initialise random numbers for repeatable results.

seed <- 123
set.seed(seed)

# Partition the full dataset into three: train (70%), validate (15%), test (15%).

nobs %>%
  sample(0.70*nobs) %T>%
  {length(.) %>% comma() %>% cat("Size of training dataset:", ., "\n")} ->
train
```

```
## Size of training dataset: 99,572
```

18 Validation Dataset

```
# Create a validation dataset of 15% of the observations.

nobs %>%
  seq_len() %>%
  setdiff(train) %>%
  sample(0.15*nobs) %T>%
  {length(.) %>% comma() %>% cat("Size of validation dataset:", ., "\n")} ->
validate
```

```
## Size of validation dataset: 21,336
```

19 Test Dataset

```
# Create a testing dataset of 15% (the remainder) of the observations.

nobs %>%
  seq_len() %>%
  setdiff(union(train, validate)) %T>%
  {length(.) %>% comma() %>% cat("Size of validation dataset:", ., "\n")} ->
test
```

```
## Size of validation dataset: 21,338
```

20 Evaluation Subsets

```
# Cache the various actual values for target and risk.

tr_target <- ds[train,][[target]]    %T>% {head(., 15) %>% print()}
```

```
##  [1] No  No  No  No  No  No  No  No  No  Yes No  Yes No  Yes No
## Levels: No Yes
```

```
tr_risk   <- ds[train,][[risk]]      %T>% {head(., 15) %>% print()}
```

```
##  [1] 0.0 0.0 0.0 0.0 0.2 0.0 0.8 0.0 0.0 1.8 0.0 5.2 0.4 1.2 0.0
```

```
va_target <- ds[validate,][[target]] %T>% {head(., 15) %>% print()}
```

```
##  [1] No  No  No  No  No  No  No  Yes No  No  No  No  No  No  No
## Levels: No Yes
```

```
va_risk   <- ds[validate,][[risk]]   %T>% {head(., 15) %>% print()}
```

```
##  [1] 0.0 0.0 0.0 0.6 0.0 0.2 0.0 4.2 0.0 0.0 0.4 0.0 0.2 0.0 0.0
```

```
te_target <- ds[test,][[target]]     %T>% {head(., 15) %>% print()}
```

```
##  [1] No  No  No  Yes No  No  No  No  No  No  No  No  No  No  Yes
## Levels: No Yes
```

```
te_risk   <- ds[test,][[risk]]       %T>% {head(., 15) %>% print()}
```

```
##  [1]  0.0  0.0  0.0 16.8  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.2
## [15]  1.8
```

21 Build Model: Decision Tree

```
# Splitting function: "anova" "poisson" "class" "exp"

mthd <- "class"

# Splitting function parameters.

prms <- list(split="information")

# Control the training.

ctrl <- rpart.control(maxdepth=5)

# Build the model

m_rp <- rpart(form, ds[train, vars], method=mthd, parms=prms, control=ctrl)
```

22 Model Generic Variables

```
# Capture the model in generic variables.

model <- m_rp
mtype <- "rpart"
mdesc <- "Decision Tree"
```

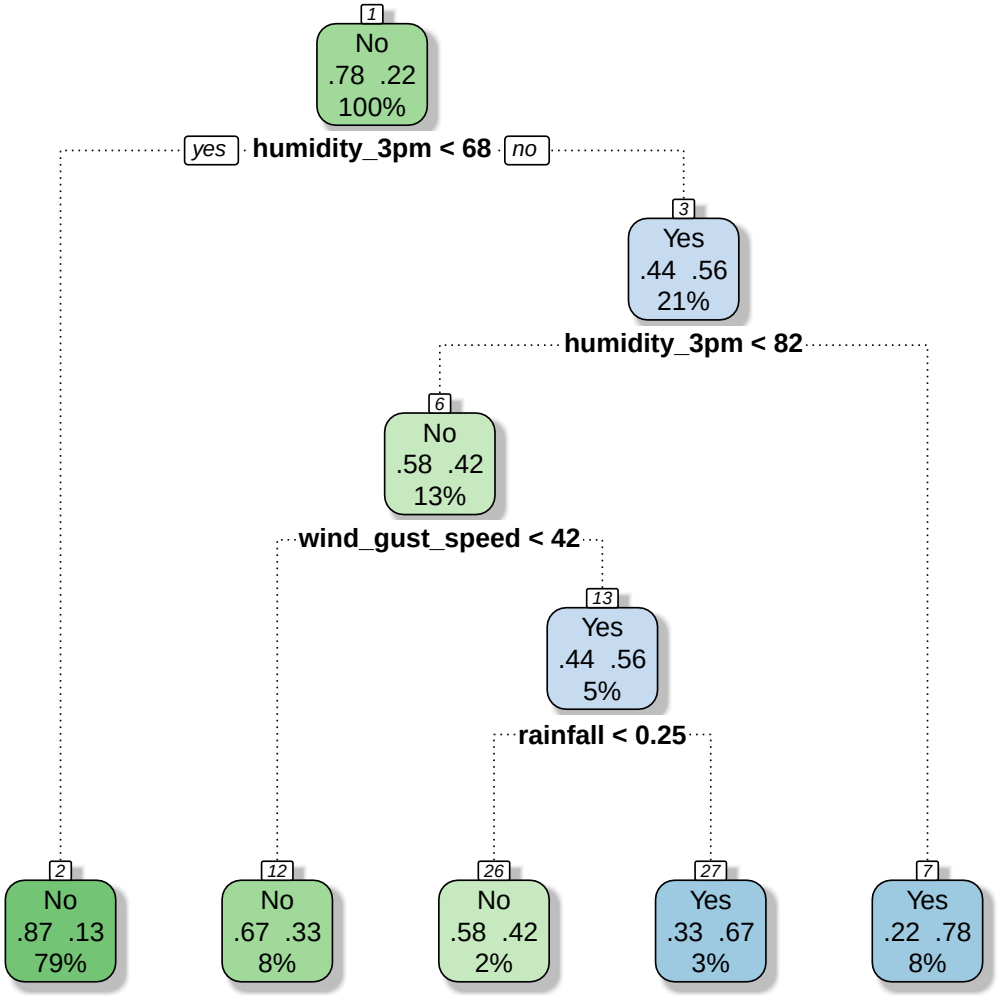
23 Review Model

```
# Basic model structure.

model
```

```
## n=99567 (5 observations deleted due to missingness)
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
##  1) root 99567 21858 No (0.7804694 0.2195306)
##    2) humidity_3pm< 67.5 78377 10073 No (0.8714802 0.1285198) *
##    3) humidity_3pm>=67.5 21190  9405 Yes (0.4438414 0.5561586)
##      6) humidity_3pm< 81.5 13295  5633 No (0.5763069 0.4236931)
##       12) wind_gust_speed< 42 7950  2633 No (0.6688050 0.3311950) *
##       13) wind_gust_speed>=42 5345  2345 Yes (0.4387278 0.5612722)
##         26) rainfall< 0.25 2274   949 No (0.5826737 0.4173263) *
##         27) rainfall>=0.25 3071  1020 Yes (0.3321394 0.6678606) *
##      7) humidity_3pm>=81.5 7895  1743 Yes (0.2207726 0.7792274) *
```

24 Visualise the Model

```
# Visually expose the discovered knowledge.

fancyRpartPlot(model)
```



25 Summary of Model

```
# Complete model build summary.

summary(model)
```

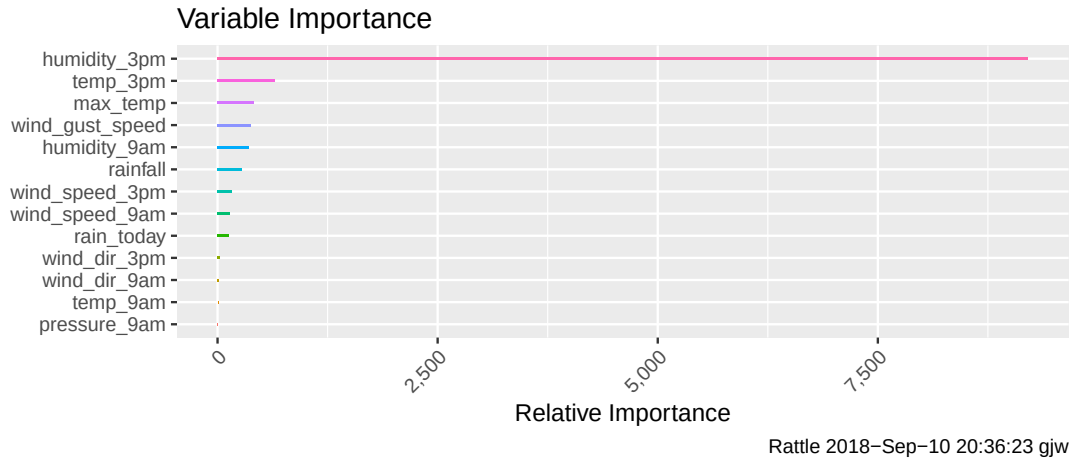
```
## Call:
## rpart(formula = form, data = ds[train, vars], method = mthd,
##     parms = prms, control = ctrl)
##   n=99567 (5 observations deleted due to missingness)
##
##           CP nsplit rel error    xerror        xstd
## 1 0.10888462      0 1.0000000 1.0000000 0.005975479
## 2 0.09282643      1 0.8911154 0.9077683 0.005766627
## 3 0.02996615      2 0.7982890 0.8016744 0.005497428
## 4 0.01720194      3 0.7683228 0.7739500 0.005421444
## 5 0.01000000      4 0.7511209 0.7610028 0.005385102
##
## Variable importance
##    humidity_3pm        temp_3pm        max_temp wind_gust_speed
##              78               6               4               3
##    humidity_9am        rainfall  wind_speed_3pm  wind_speed_9am
##               3               2               1               1
##      rain_today
##               1
##
## Node number 1: 99567 observations,    complexity param=0.1088846
##   predicted class=No   expected loss=0.2195306  P(node) =1
##     class counts: 77709 21858
##    probabilities: 0.780 0.220
##   left son=2 (78377 obs) right son=3 (21190 obs)
##   Primary splits:
##       humidity_3pm < 67.5    to the left,  improve=7867.246, (2512 missing)
##       rainfall     < 0.35    to the left,  improve=4776.094, (998 missing)
##       rain_today   splits as  LR,          improve=4550.271, (998 missing)
##       sunshine     < 7.75    to the right, improve=4176.126, (48248 missing)
##       cloud_3pm    < 5.5     to the left,  improve=3528.719, (40795 missing)
##   Surrogate splits:
##       temp_3pm     < 11.45   to the right, agree=0.799, adj=0.072, (601 split)
##       max_temp     < 11.95   to the right, agree=0.792, adj=0.042, (1736 split)
##       humidity_9am < 91.5    to the left,  agree=0.790, adj=0.032, (113 split)
##       rainfall     < 12.45   to the left,  agree=0.786, adj=0.011, (60 split)
##
## Node number 2: 78377 observations
##   predicted class=No   expected loss=0.1285198  P(node) =0.7871785
##     class counts: 68304 10073
##    probabilities: 0.871 0.129
##
## Node number 3: 21190 observations,    complexity param=0.09282643
```

```
##   predicted class=Yes  expected loss=0.4438414  P(node) =0.2128215
##     class counts:  9405 11785
##    probabilities: 0.444 0.556
##   left son=6 (13295 obs) right son=7 (7895 obs)
##   Primary splits:
##       humidity_3pm    < 81.5    to the left,  improve=1340.2430, (159 missing)
##       rainfall        < 2.25    to the left,  improve= 773.1974, (250 missing)
##       rain_today      splits as  LR,          improve= 730.2194, (250 missing)
##       wind_gust_speed < 42      to the left,  improve= 553.1646, (1701 missing)
##       pressure_9am    < 1014.85 to the right, improve= 483.8505, (2413 missing)
##   Surrogate splits:
##       temp_3pm     < 10.25   to the right, agree=0.655, adj=0.068, (48 split)
##       max_temp     < 11.15   to the right, agree=0.649, adj=0.049, (94 split)
##       humidity_9am < 94.5    to the left,  agree=0.645, adj=0.041, (15 split)
##       rainfall     < 20.55   to the left,  agree=0.634, adj=0.009, (2 split)
##       temp_9am     < 0.75    to the right, agree=0.633, adj=0.008, (0 split)
##
## Node number 6: 13295 observations,    complexity param=0.02996615
##   predicted class=No   expected loss=0.4236931  P(node) =0.1335282
##     class counts:  7662  5633
##    probabilities: 0.576 0.424
##   left son=12 (7950 obs) right son=13 (5345 obs)
##   Primary splits:
##       wind_gust_speed < 42      to the left,  improve=375.4905, (1149 missing)
##       rainfall        < 2.25    to the left,  improve=370.9427, (137 missing)
##       rain_today      splits as  LR,          improve=345.9041, (137 missing)
##       pressure_3pm    < 1013.95 to the right, improve=269.5676, (1130 missing)
##       pressure_9am    < 1014.85 to the right, improve=266.6571, (1148 missing)
##   Surrogate splits:
##       wind_speed_3pm < 23      to the left,  agree=0.772, adj=0.446, (1047 split)
##       wind_speed_9am < 18      to the left,  agree=0.746, adj=0.383, (12 split)
##       rainfall       < 5.3     to the left,  agree=0.604, adj=0.039, (90 split)
##       humidity_9am   < 66.5    to the right, agree=0.602, adj=0.035, (0 split)
##       pressure_9am   < 1013.45 to the right, agree=0.598, adj=0.025, (0 split)
##
## Node number 7: 7895 observations
##   predicted class=Yes  expected loss=0.2207726  P(node) =0.07929334
##     class counts:  1743  6152
##    probabilities: 0.221 0.779
##
## Node number 12: 7950 observations
##   predicted class=No   expected loss=0.331195  P(node) =0.07984573
##     class counts:  5317  2633
##    probabilities: 0.669 0.331
##
## Node number 13: 5345 observations,    complexity param=0.01720194
##   predicted class=Yes  expected loss=0.4387278  P(node) =0.05368244
##     class counts:  2345  3000
```

```
##    probabilities: 0.439 0.561
##   left son=26 (2274 obs) right son=27 (3071 obs)
##   Primary splits:
##       rainfall        < 0.25    to the left,  improve=166.35160, (52 missing)
##       rain_today      splits as  LR, improve=141.16380, (52 missing)
##       sunshine        < 9.05    to the right, improve= 96.47547, (2666 missing)
##       wind_gust_speed < 55      to the left,  improve= 90.68309, (338 missing)
##       wind_dir_3pm    splits as  LLLRLLRRLLLRRRRR, improve= 87.96325, (36 missing)
##   Surrogate splits:
##       rain_today   splits as  LR, agree=0.903, adj=0.773, (0 split)
##       humidity_9am < 76.5    to the left,  agree=0.675, adj=0.237, (40 split)
##       wind_dir_3pm splits as  LLLLLLLRRRRRRRRR, agree=0.641, adj=0.156, (11 split)
##       wind_dir_9am splits as  LLRLLLRRRRRRRRRR, agree=0.627, adj=0.124, (0 split)
##       max_temp     < 21.15   to the right, agree=0.619, adj=0.105, (1 split)
##
## Node number 26: 2274 observations
##   predicted class=No   expected loss=0.4173263  P(node) =0.02283889
##     class counts:  1325   949
##    probabilities: 0.583 0.417
##
## Node number 27: 3071 observations
##   predicted class=Yes  expected loss=0.3321394  P(node) =0.03084355
##     class counts:  1020  2051
##    probabilities: 0.332 0.668
```

26 Variable Importance

```
# Review which importance of the variables.

ggVarImp(model)
```



27 Model Predictions on Validation

```
# Predict on validation dataset to judge performance.

model %>%
  predict(newdata=ds[validate, vars], type="class") %>%
  set_names(NULL) %T>%
  {head(., 20) %>% print()} ->
va_class
```

```
##  [1] No  No  No  Yes No  No  No  No  No  No  No  No  No  No  No  No  No
## [18] No  No  No
## Levels: No Yes
```

```
model %>%
  predict(newdata=ds[validate, vars], type="prob") %>%
  .[,2] %>%
  set_names(NULL) %>%
  round(2) %T>%
  {head(., 20) %>% print()} ->
va_prob
```

```
##  [1] 0.13 0.13 0.13 0.67 0.13 0.33 0.13 0.13 0.13 0.13 0.13 0.13 0.33 0.13
## [15] 0.13 0.13 0.13 0.13 0.13 0.13
```

28 Overall Accuracy and Error

```
# Overall accuracy and error.

sum(va_class == va_target) %>%
  divide_by(length(va_target)) %T>%
  {
    percent(.) %>%
      sprintf("Overall accuracy = %s\n", .) %>%
      cat()
  } ->
va_acc
```

```
## Overall accuracy = 83.4%
```

```
sum(va_class != va_target) %>%
  divide_by(length(va_target)) %T>%
  {
    percent(.) %>%
      sprintf("Overall error = %s\n", .) %>%
      cat()
  } ->
va_err
```

```
## Overall error = 16.6%
```

29 Confusion Matrix

```
# Basic comparison of prediction/actual as a confusion matrix.

table(va_target, va_class, useNA="ifany", dnn=c("Actual", "Predicted"))
```

```
##        Predicted
## Actual    No   Yes
##    No  16066   616
##    Yes  2923  1731
```

```
# Comparison as percentages of all observations.

errorMatrix(va_target, va_class) %T>%
  print() ->
va_matrix
```

```
##        Predicted
## Actual   No Yes Error
##    No  75.3 2.9   3.7
##    Yes 13.7 8.1  62.8
```

```
# Error rate and average of the class error rate.

va_matrix %>%
  diag() %>%
  sum(na.rm=TRUE) %>%
  subtract(100, .) %>%
  sprintf("Overall error percentage = %s%%\n", .) %>%
  cat()
```

```
## Overall error percentage = 16.6%
```

```
va_matrix[,"Error"] %>%
  mean(na.rm=TRUE) %>%
  sprintf("Averaged class error percentage = %s%%\n", .) %>%
  cat()
```

```
## Averaged class error percentage = 33.25%
```

30 Recall, Precision, F-Score

```
# Other performance metrics: recall, precision, and the F-score.

va_rec <- (va_matrix[2,2]/(va_matrix[2,2]+va_matrix[2,1])) %T>%
  {percent(.) %>% sprintf("Recall = %s\n", .) %>% cat()}
```

```
## Recall = 37.2%
```

```
va_pre <- (va_matrix[2,2]/(va_matrix[2,2]+va_matrix[1,2])) %T>%
  {percent(.) %>% sprintf("Precision = %s\n", .) %>% cat()}
```

```
## Precision = 73.6%
```

```
va_fsc <- ((2 * va_pre * va_rec)/(va_rec + va_pre))  %T>%
  {sprintf("F-Score = %.3f\n", .) %>% cat()}
```

```
## F-Score = 0.494
```

31 ROC Curve

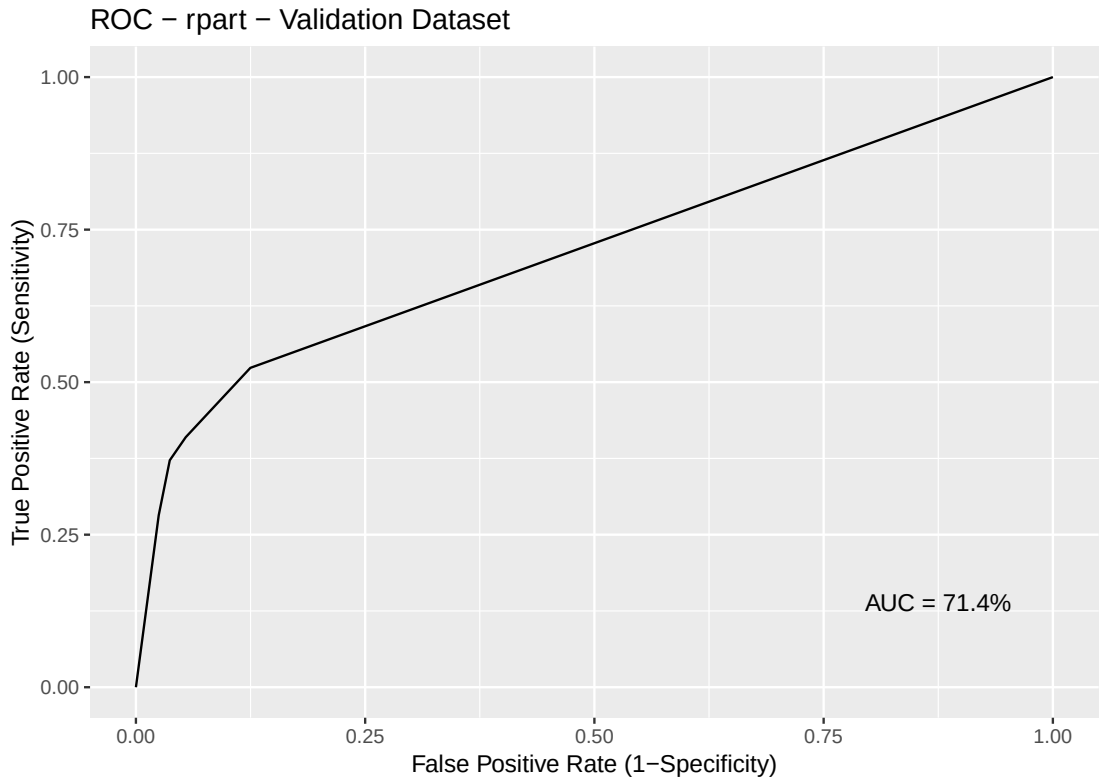
```
# Calculate the area under the curve (AUC).

va_prob %>%
  prediction(va_target) %>%
  performance("auc") %>%
  attr("y.values") %>%
  .[[1]] %T>%
  {
    percent(.) %>%
    sprintf("Percentage area under the ROC curve = %s\n", .) %>%
    cat()
  } ->
va_auc
```

```
## Percentage area under the ROC curve = 71.4%
```

```
# Calculate measures required to plot the ROC Curve.

va_prob %>%
  prediction(va_target) %>%
  performance("tpr", "fpr") ->
va_rates
```

32 ROC Curve Plot

```
# Plot the ROC Curve.

data_frame(tpr=attr(va_rates, "y.values")[[1]],
           fpr=attr(va_rates, "x.values")[[1]]) %>%
  ggplot(aes(fpr, tpr)) +
  geom_line() +
  annotate("text", x=0.875, y=0.125, vjust=0,
           label=paste("AUC =", percent(va_auc))) +
  labs(title="ROC - " %s+% mtype %s+% " - Validation Dataset",
       x="False Positive Rate (1-Specificity)",
       y="True Positive Rate (Sensitivity)")
```



33 Risk Chart

```
# Risk chart.

riskchart(va_prob, va_target, va_risk) +
  labs(title="Risk Chart - " %s+%
         mtype %s+%
         " - Validation Dataset") +
  theme(plot.title=element_text(size=14))
```