Everest

Victor M. Uribe

2022-09-19

READ ME

To do List: (10/27/22 - Last Worked On)

- create some new features
- check correlation between variables
- verify if the assumptions have been violated or not

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(rsample)
ggpairs <- GGally::ggpairs
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
select <- dplyr::select
```

```
members <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/20
```

```
## Rows: 76519 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (10): expedition_id, member_id, peak_id, peak_name, season, sex, citizen...
## dbl  (5): year, age, highpoint_metres, death_height_metres, injury_height_me...
## lgl  (6): hired, success, solo, oxygen_used, died, injured
##
## 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.
```

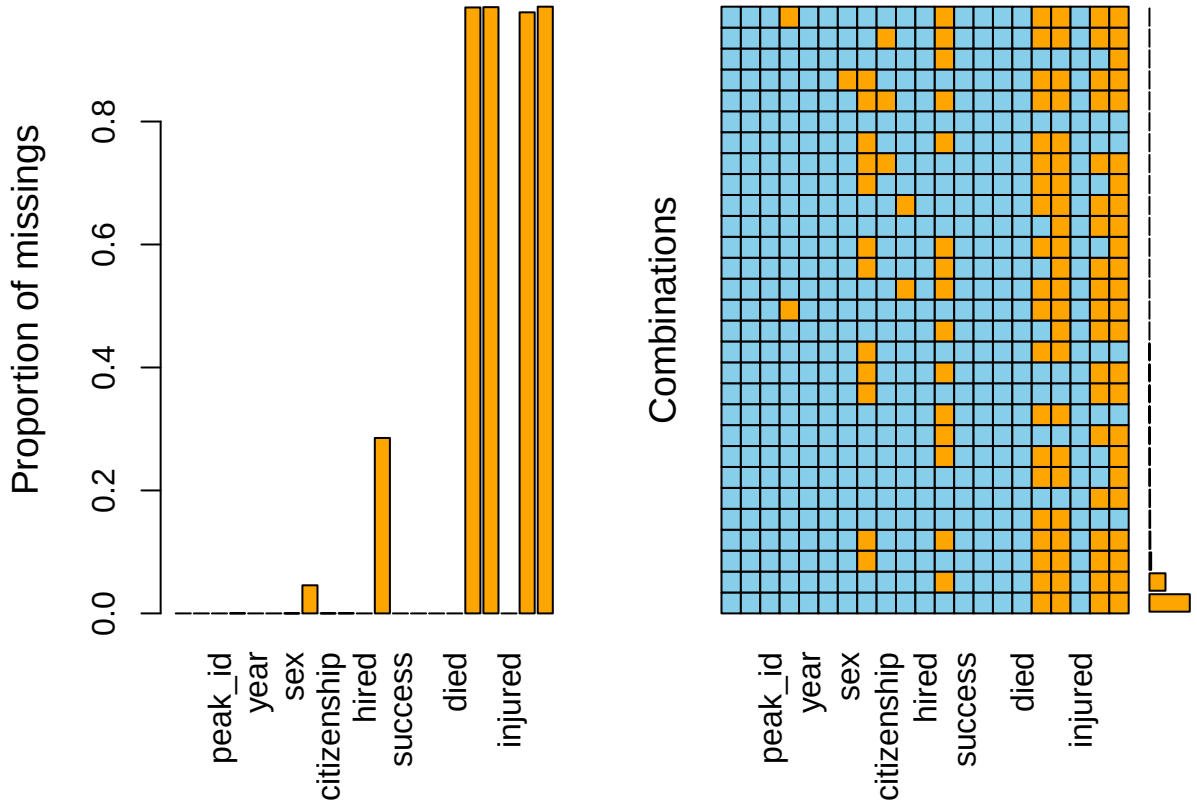
```
glimpse(members)
```

```
## Rows: 76,519
## Columns: 21
## $ expedition_id        <chr> "AMAD78301", "AMAD78301", "AMAD78301", "AMAD78301~
## $ member_id            <chr> "AMAD78301-01", "AMAD78301-02", "AMAD78301-03", "~
## $ peak_id              <chr> "AMAD", "AMAD", "AMAD", "AMAD", "AMAD", "AMAD", "~
```

```
## $ peak_name            <chr> "Ama Dablam", "Ama Dablam", "Ama Dablam", "Ama Da~
## $ year                 <dbl> 1978, 1978, 1978, 1978, 1978, 1978, 1978, 1978, 1~
## $ season               <chr> "Autumn", "Autumn", "Autumn", "Autumn", "Autumn",~
## $ sex                  <chr> "M", "M", "M", "M", "M", "M", "M", "M", "M", "M",~
## $ age                  <dbl> 40, 41, 27, 40, 34, 25, 41, 29, 35, 37, 23, 44, 2~
## $ citizenship          <chr> "France", "France", "France", "France", "France",~
## $ expedition_role      <chr> "Leader", "Deputy Leader", "Climber", "Exp Doctor~
## $ hired                <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, ~
## $ highpoint_metres     <dbl> NA, 6000, NA, 6000, NA, 6000, 6000, 6000, NA, 681~
## $ success              <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, ~
## $ solo                 <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, ~
## $ oxygen_used          <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, ~
## $ died                 <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, ~
## $ death_cause          <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ death_height_metres  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ injured              <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, ~
## $ injury_type          <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ injury_height_metres <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
```

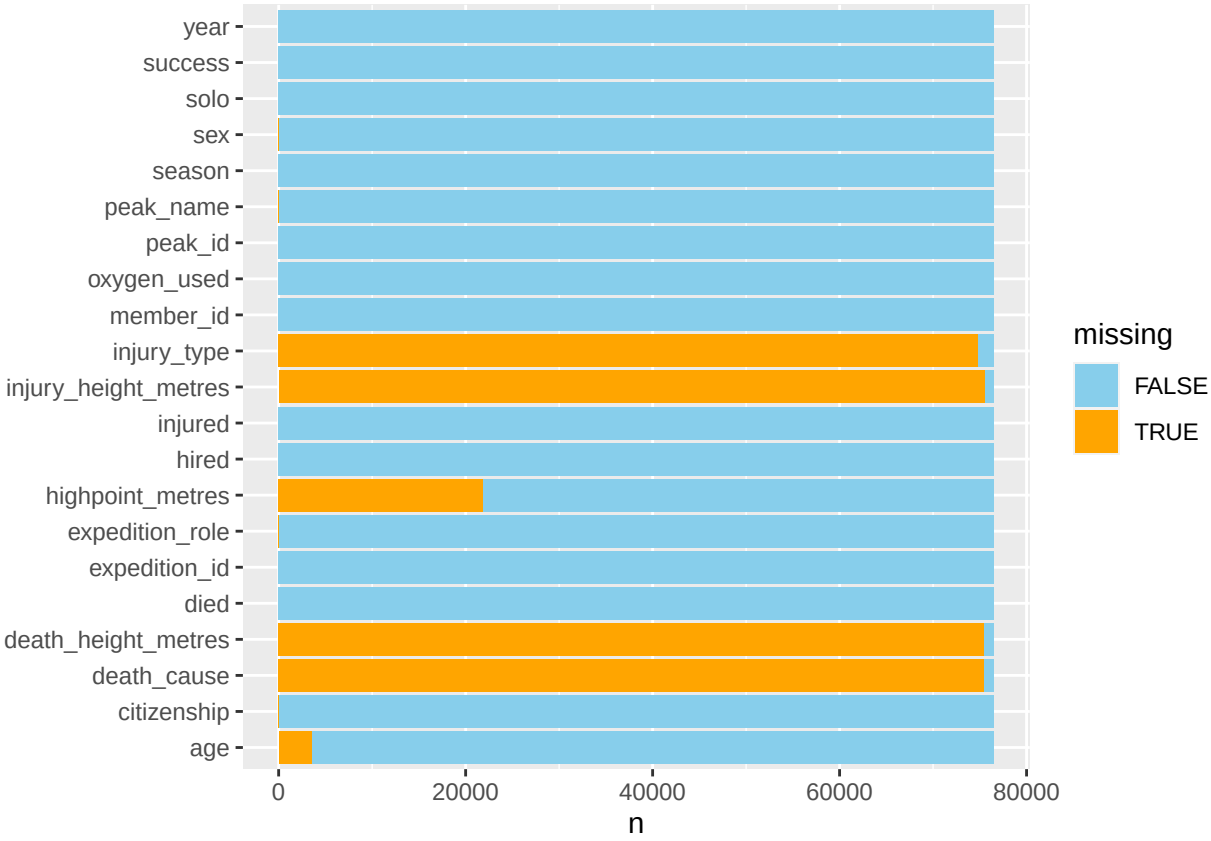
```
attach(members)
```

```
VIM::aggr(members, col = c("skyblue","orange"))
```




```
missvalues_visual <-
    members  %>%
      summarise_all(list(~is.na(.)))%>%
      pivot_longer(everything(),
                   names_to = "variables", values_to="missing") %>%
      count(variables, missing) %>%
```

```
      ggplot(aes(y=variables,x=n,fill=missing))+
      geom_col()+
      scale_fill_manual(values=c("skyblue","orange"))+
      theme(axis.title.y=element_blank())
missvalues_visual
```



So most of the data that is giving issues is composed of mostly na

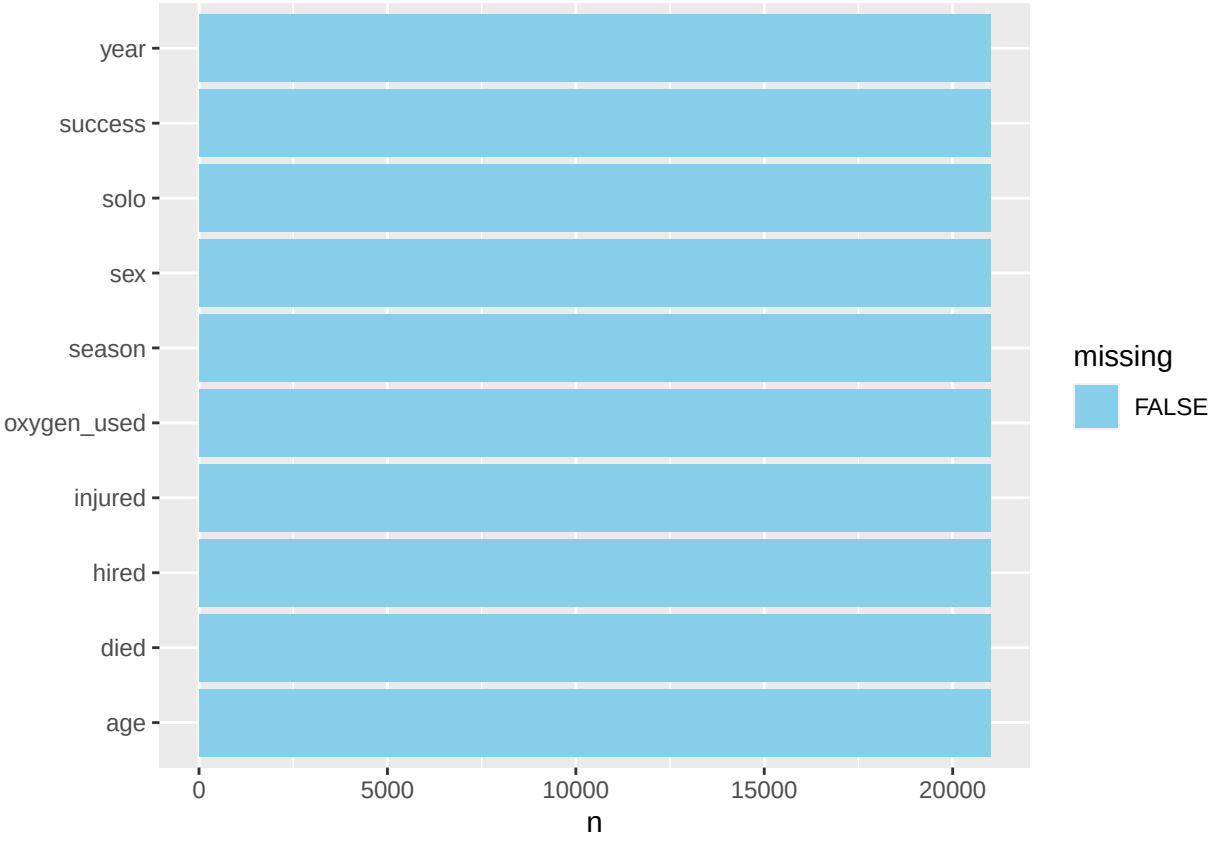
Data cleaning

```
everest <- members %>% # should have an age
  filter(age != 'NA' & peak_name == "Everest") %>%
  select(-c(peak_id,peak_name,expedition_id,member_id, death_height_metres,
            injury_height_metres, highpoint_metres, death_cause,
            citizenship, expedition_role, injury_type)) %>%
  mutate(
    #death_height_metres = ifelse(death_height_metres == NA, 0, death_height_metres), # issue with lev
    #injury_height_metres = ifelse(injury_height_metres == NA, 0, injury_height_metres), # gives leveli
    #highpoint_metres = ifelse(highpoint_metres == 'NA', 0, highpoint_metres), # thows off the p values
    season = factor(season),
    sex = factor(sex),
    #citizenship = factor(citizenship), # not significant after testing
    #expedition_role = factor(expedition_role), # not significant after testing
    hired = factor(hired),
    success = factor(success), # value being predicted
    solo = factor(solo),
```

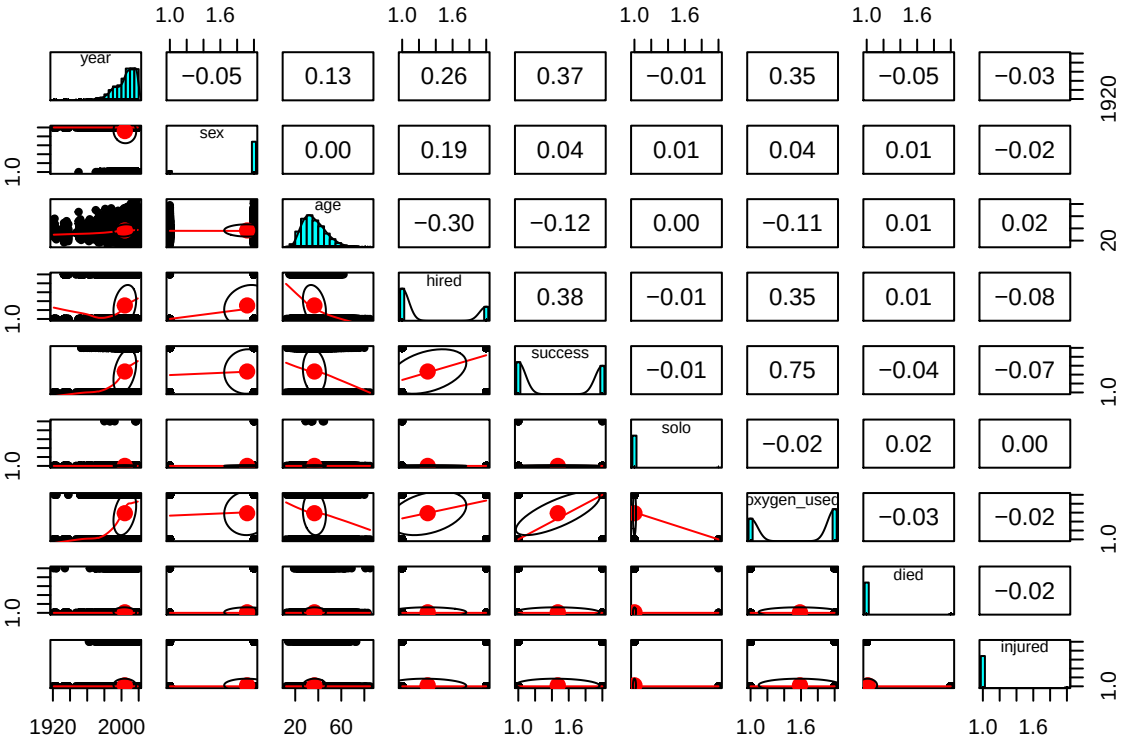
```
    oxygen_used = factor(oxygen_used),
    died = factor(died),
    #death_cause = factor(death_cause), # issue
    injured = factor(injured)#,
    #injury_type = factor(injury_type) # issue with levels
  )
glimpse(everest)
```

```
## Rows: 20,997
## Columns: 10
## $ year        <dbl> 1963, 1963, 1963, 1963, 1963, 1963, 1963, 1963, 1963, 1963~
## $ season      <fct> Spring, Spring, Spring, Spring, Spring, Spring, Spring, Sp~
## $ sex         <fct> M, M, M, M, M, M, M, M, M, M, M, M, M, M, M, M, M, M, M, M~
## $ age         <dbl> 36, 31, 27, 26, 26, 29, 44, 37, 32, 26, 34, 42, 35, 23, 27~
## $ hired       <fct> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FA~
## $ success     <fct> FALSE, TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, TRU~
## $ solo        <fct> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FA~
## $ oxygen_used <fct> TRUE, TRUE, FALSE, TRUE, TRUE, FALSE, TRUE, TRUE, TRUE, TR~
## $ died        <fct> FALSE, FALSE, TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, FAL~
## $ injured     <fct> FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, FALSE, FALSE, FAL~
```

```
missvalues_visual2 <-
    everest  %>%
      summarise_all(list(~is.na(.)))%>%
      pivot_longer(everything(),
                   names_to = "variables", values_to="missing") %>%
      count(variables, missing) %>%
      ggplot(aes(y=variables,x=n,fill=missing))+
      geom_col()+
      scale_fill_manual(values=c("skyblue","orange"))+
      theme(axis.title.y=element_blank())
missvalues_visual2
```




```
psych::pairs.panels(everest[,c(-2)])
```



So we can still use accuracy as a measure of the models predictability

Assumptions

Logistic Regression Assumptions:

- Response variable is binary or dichotomous
- No multicollinearity among the predictor variables
- Linear relationship of independent variables to log odds
- large sample size
- Problem with extreme outliers
- independent observations

Training and Testing data

```
set.seed(5302)
split <- initial_split(everest, prop = .70)

train_data <- training(split)
test_data <- testing(split)
```

fiting models

```
glm(success ~.,data = train_data, family = "binomial") %>% summary()
```

```
##
## Call:
## glm(formula = success ~ ., family = "binomial", data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3858  -0.2417  -0.0912   0.5803   3.4021
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -96.695817   5.629909 -17.175  < 2e-16 ***
## year              0.046456   0.002830  16.415  < 2e-16 ***
## seasonSpring      0.708208   0.111437   6.355 2.08e-10 ***
## seasonSummer     -0.136322   0.491471  -0.277   0.7815
## seasonWinter     -0.835367   0.378939  -2.204   0.0275 *
## sexM              0.030782   0.097500   0.316   0.7522
## age              -0.020252   0.002876  -7.043 1.88e-12 ***
## hiredTRUE         0.800386   0.063701  12.565  < 2e-16 ***
## soloTRUE          4.378202   1.055927   4.146 3.38e-05 ***
## oxygen_usedTRUE   4.552426   0.087353  52.116  < 2e-16 ***
## diedTRUE          0.013819   0.220821   0.063   0.9501
## injuredTRUE      -0.797349   0.163508  -4.877 1.08e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 20317.4  on 14696  degrees of freedom
## Residual deviance:  9433.5  on 14685  degrees of freedom
## AIC: 9457.5
##
## Number of Fisher Scoring iterations: 6
```

```
sig_mod <- glm(success ~., data = train_data, family = "binomial") %>% MASS::stepAIC()
```

```
## Start:  AIC=9457.45
## success ~ year + season + sex + age + hired + solo + oxygen_used +
##     died + injured
##
##               Df Deviance     AIC
## - died         1   9433.5  9455.5
## - sex          1   9433.6  9455.6
## <none>             9433.5  9457.5
## - solo         1   9444.7  9466.7
## - injured      1   9457.0  9479.0
## - age          1   9483.2  9505.2
## - season       3   9492.6  9510.6
## - hired        1   9595.3  9617.3
## - year         1   9725.7  9747.7
## - oxygen_used  1  16042.5 16064.5
##
## Step:  AIC=9455.46
## success ~ year + season + sex + age + hired + solo + oxygen_used +
##     injured
##
##               Df Deviance     AIC
## - sex          1   9433.6  9453.6
## <none>             9433.5  9455.5
## - solo         1   9444.7  9464.7
## - injured      1   9457.0  9477.0
## - age          1   9483.3  9503.3
## - season       3   9492.6  9508.6
## - hired        1   9595.4  9615.4
## - year         1   9726.4  9746.4
## - oxygen_used  1  16048.3 16068.3
##
## Step:  AIC=9453.56
## success ~ year + season + age + hired + solo + oxygen_used +
##     injured
##
##               Df Deviance     AIC
## <none>             9433.6  9453.6
## - solo         1   9444.8  9462.8
## - injured      1   9457.2  9475.2
## - age          1   9483.4  9501.4
## - season       3   9492.7  9506.7
## - hired        1   9607.1  9625.1
## - year         1   9729.9  9747.9
## - oxygen_used  1  16049.9 16067.9
```

```
# Predict on test
p <- predict(sig_mod, newdata = test_data, type = "response")
```

```r
# If p exceeds threshold of 0.5, 1 else 0
yes_no <- ifelse(p > 0.5, 1, 0)

# Convert to factor: p_class
p_class <- factor(ifelse(yes_no == 1, TRUE, FALSE))

# Create confusion matrix
caret::confusionMatrix(p_class, test_data[["success"]])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  2670  122
##      TRUE    658 2850
##
##                Accuracy : 0.8762
##                  95% CI : (0.8678, 0.8842)
##     No Information Rate : 0.5283
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.754
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8023
##             Specificity : 0.9590
##          Pos Pred Value : 0.9563
##          Neg Pred Value : 0.8124
##              Prevalence : 0.5283
##          Detection Rate : 0.4238
##    Detection Prevalence : 0.4432
##       Balanced Accuracy : 0.8806
##
##        'Positive' Class : FALSE
##
```

Accuracy : 0.8765