Forest_Fire_logistic_regression

Bowei Zhang

2023-03-27

Data Import and Preprocessing

import data

```
df_raw = read_csv('forest_fire.csv')
```

```
## Rows: 517 Columns: 14
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): month, day
## dbl (12): area, X, Y, FFMC, DMC, DC, ISI, temp, RH, wind, rain, id
##
## 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.
```

```
head(df_raw)
```

```
## # A tibble: 6 x 14
##    area     X     Y month day    FFMC   DMC    DC   ISI  temp    RH  wind  rain
##   <dbl> <dbl> <dbl> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1   0.1     7     5 mar   fri    86.2  26.2  94.3   5.1   8.2    51   6.7   0
## 2   0.1     7     4 oct   tue    90.6  35.4 669.    6.7  18      33   0.9   0
## 3   0.1     7     4 oct   sat    90.6  43.7 687.    6.7  14.6    33   1.3   0
## 4   0.1     8     6 mar   fri    91.7  33.3  77.5   9     8.3    97   4     0.2
## 5   0.1     8     6 mar   sun    89.3  51.3 102.    9.6  11.4    99   1.8   0
## 6   0.1     8     6 aug   sun    92.3  85.3 488    14.7  22.2    29   5.4   0
## # ... with 1 more variable: id <dbl>
```

preprocess

month and day

```
months <- c("jan", "feb", "mar", "apr", "may", "jun", "jul", "aug", "sep", "oct", "nov", "dec")
days <- c("mon", "tue", "wed", "thu", "fri", "sat", "sun")

df <- df_raw %>%
  mutate(month = case_when(
    month %in% months ~ match(month, months)
  )) %>%
  mutate(day = case_when(
    day %in% days ~ match(day, days)
  ))

df$month <- factor(df$month,
                   levels = 1:12,
                   labels = months)
df$day <- factor(df$day,
                 levels = 1:7,
                 labels = days)
```

```
df <- df %>%
  mutate(area_size = case_when(
    area == 0.1 ~ 0,
    TRUE ~ 1
  ))
```

model fitting

```
mod.fit.full <- logistf(area_size ~ X + Y + month + day + temp + RH + wind + rain,
               data = df,
               family = binomial)
backward(mod.fit.full, slstay = 0.20)
```

```
## Step  0 : starting model
## Step  1 : removed  day  (P= 0.9183893 )
## Step  2 : removed  rain  (P= 0.9764783 )
## Step  3 : removed  RH  (P= 0.7457029 )
## Step  4 : removed  Y  (P= 0.6493052 )
## Step  5 : removed  wind  (P= 0.245984 )
```

```
## logistf(formula = area_size ~ X + month + temp, data = df, family = binomial)
## Model fitted by Penalized ML
## Confidence intervals and p-values by Profile Likelihood
##
## Coefficients:
## (Intercept)           X    monthfeb    monthmar    monthapr    monthmay
## -1.98988120  0.07035215  1.32101470  0.63969417  0.99891912  1.17180159
##    monthjun    monthjul    monthaug    monthsep    monthoct    monthnov
##  0.78285793  1.16594726  1.11693453  1.31432971  0.38279110  0.09355447
##    monthdec        temp
##  4.47496944  0.03183064
##
## Likelihood ratio test=26.87089 on 13 df, p=0.01295737, n=517
```

```
mod.fit <- logistf(formula = area_size ~ X + month + temp,
                   data = df,
                   family = binomial)
summary(mod.fit)
```

```
## logistf(formula = area_size ~ X + month + temp, data = df, family = binomial)
##
## Model fitted by Penalized ML
## Coefficients:
##                    coef   se(coef)   lower 0.95  upper 0.95       Chisq
## (Intercept) -1.98988120 1.55836660 -6.924144670  0.54301896 2.276357887
## X            0.07035215 0.03935827 -0.006605016  0.14806907 3.208922529
## monthfeb     1.32101470 1.61486532 -1.354502797  6.29957503 0.828619873
## monthmar     0.63969417 1.58543526 -1.962833537  5.59483003 0.183905882
## monthapr     0.99891912 1.68485842 -1.858134314  6.03417388 0.404133076
## monthmay     1.17180159 1.94398002 -2.372716038  6.44629277 0.396018798
## monthjun     0.78285793 1.65770688 -2.001064590  5.79596910 0.251709663
## monthjul     1.16594726 1.63032927 -1.547699739  6.15695175 0.615396949
## monthaug     1.11693453 1.59513918 -1.509226012  6.07979189 0.596566199
## monthsep     1.31432971 1.58684144 -1.290882559  6.27065704 0.863219278
## monthoct     0.38279110 1.66042149 -2.415817873  5.39740138 0.056476261
## monthnov     0.09355447 2.25855008 -5.387344636  5.59846562 0.001715567
## monthdec     4.47496944 2.12305277  1.051932792 10.31685482 7.068089513
## temp         0.03183064 0.02094624 -0.009071621  0.07327184 2.323734305
##                       p method
## (Intercept) 0.131360029      2
## X           0.073237697      2
## monthfeb    0.362671911      2
## monthmar    0.668037854      2
## monthapr    0.524962447      2
## monthmay    0.529152511      2
## monthjun    0.615873815      2
## monthjul    0.432762969      2
## monthaug    0.439891184      2
## monthsep    0.352839365      2
## monthoct    0.812154626      2
## monthnov    0.966961544      2
## monthdec    0.007846888      2
## temp        0.127413820      2
##
## Method: 1-Wald, 2-Profile penalized log-likelihood, 3-None
##
## Likelihood ratio test=26.87089 on 13 df, p=0.01295737, n=517
## Wald test = 20.84932 on 13 df, p = 0.07596292
```

```
mod.fit.full2 <- logistf(formula = area_size ~ FFMC + DMC + DC + ISI,
                   data = df,
                   family = binomial)
backward(mod.fit.full2, slstay = 0.20)
```

```
## Step  0 : starting model
## Step  1 : removed  ISI  (P= 0.8263427 )
## Step  2 : removed  DMC  (P= 0.6828909 )
## Step  3 : removed  FFMC  (P= 0.3426695 )

## logistf(formula = area_size ~ DC, data = df, pl = FALSE, family = binomial)
## Model fitted by Penalized ML
## Confidence intervals and p-values by Wald
##
## Coefficients:
##   (Intercept)            DC
## -0.3349555602  0.0007599967
##
## Likelihood ratio test=4.580487 on 1 df, p=0.03233803, n=517
```

```
mod.fit.2 <- glm(formula = area_size ~ DC,
                 data = df,
                 family = binomial)
summary(mod.fit.2)
```

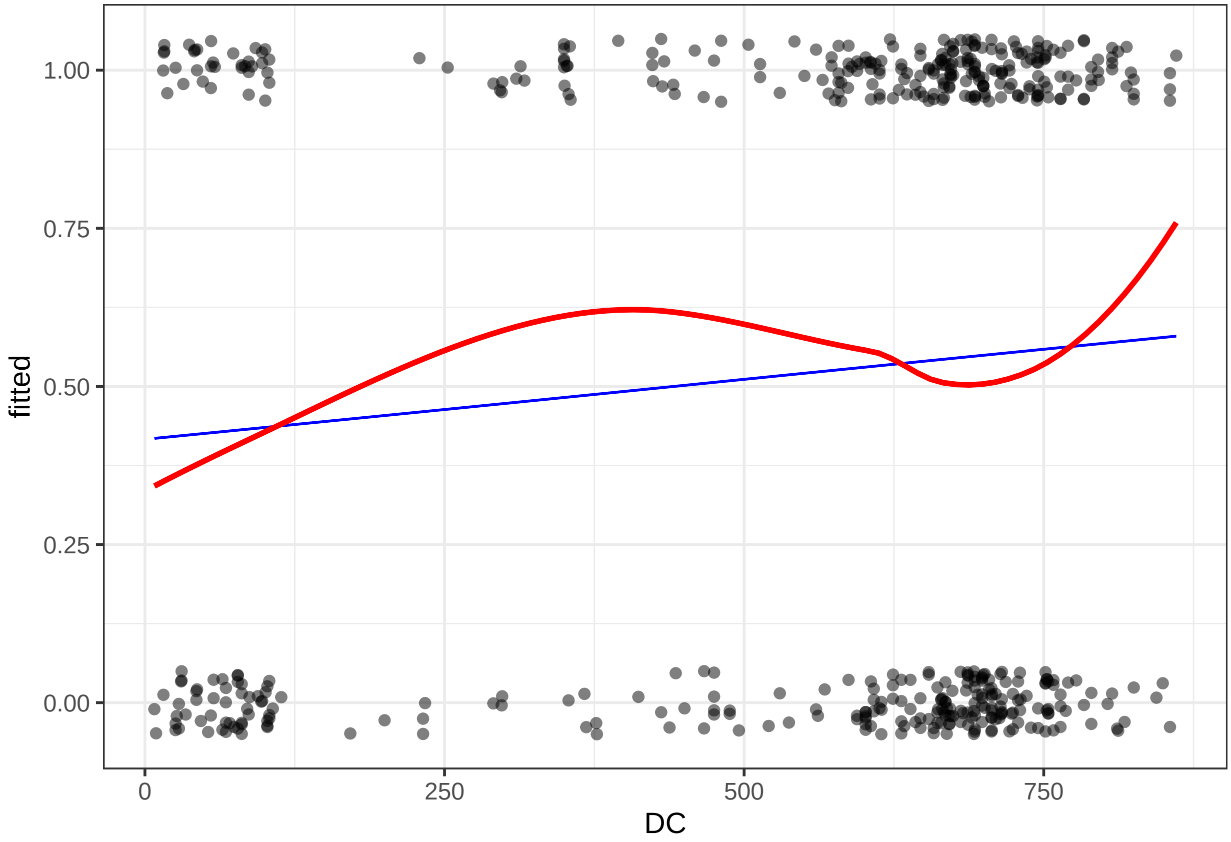
```
##
## Call:
## glm(formula = area_size ~ DC, family = binomial, data = df)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.314  -1.247   1.060   1.108   1.319
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.3376981  0.2153865  -1.568   0.1169
## DC           0.0007648  0.0003581   2.136   0.0327 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 715.86  on 516  degrees of freedom
## Residual deviance: 711.26  on 515  degrees of freedom
## AIC: 715.26
##
## Number of Fisher Scoring iterations: 3
```

Visualization

```
# Create fitted df
new.df <- data.frame(
  DC = with(df,
            seq(min(DC),
                max(DC),
                by = 0.1),
            )
)
new.df$fitted <- predict(mod.fit.2, newdata = new.df, type = "response")

# ggplot2
ggplot() +
  geom_line(data = new.df,
            aes(x = DC,
                y = fitted,
                group = 1),
            col = 'blue') +
  geom_jitter(data = df,
             aes(x = DC,
                 y = area_size),
             height = 0.05,
             width = 0.05,
             alpha = 0.5) +
  geom_smooth(data = df,
              aes(x = DC,
                  y = area_size),
              col = 'red',
              se = FALSE) +
  theme_bw()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

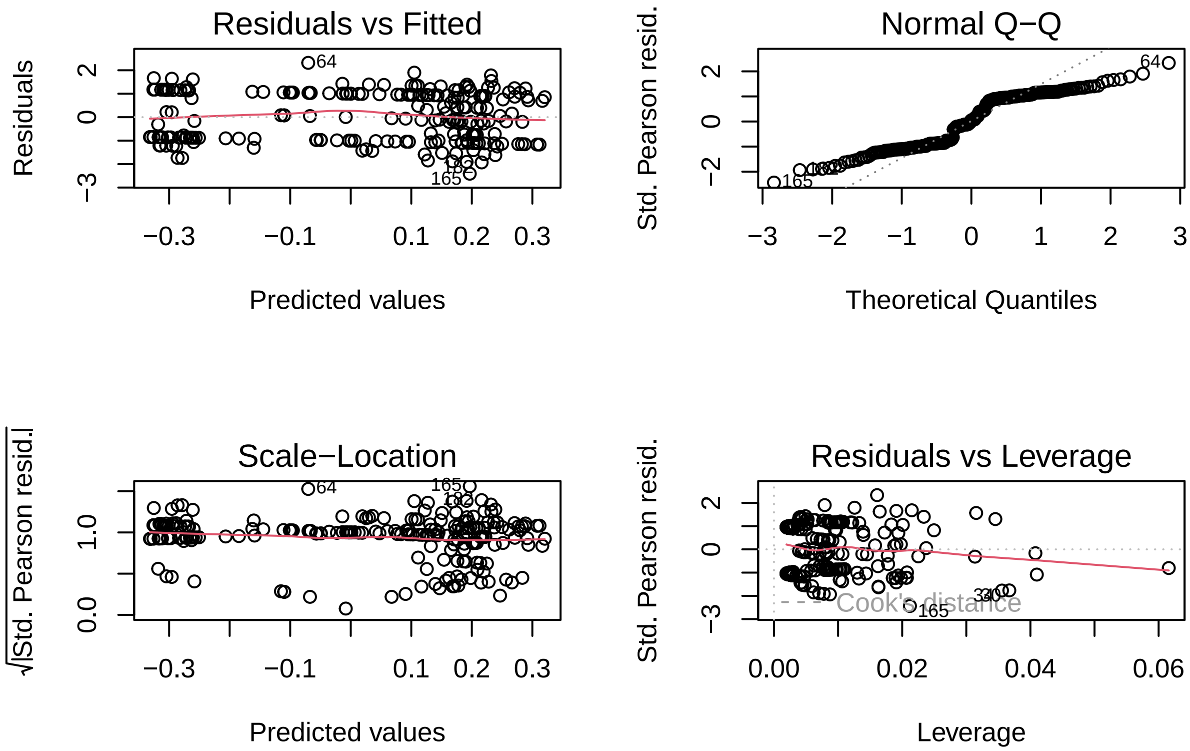


# Diagnostic

```r
# aggregate data
ag.df <- aggregate(area_size ~ DC, data = df, FUN = sum)
ag.df <- cbind(ag.df,
               aggregate(area ~ DC,
                         data = df,
                         FUN = length))
names(ag.df)[ncol(ag.df)] <- 'tot'
ag.df <- ag.df[,!duplicated(names(ag.df))]

# fit aggregate model
ag.mod.fit <- glm(area_size/tot ~ DC,
                  data = ag.df,
                  family = binomial,
                  weight = tot)

# plot diagnostics
par(mfrow = c(2,2))
plot(ag.mod.fit)
```



```r
generalhoslem::logitgof(df$area_size, mod.fit.2$fitted.values)
```

```
##
##  Hosmer and Lemeshow test (binary model)
##
## data:  df$area_size, mod.fit.2$fitted.values
## X-squared = 11.922, df = 8, p-value = 0.1547
```