Predictive Modeling and Simulations of Forest Fire Area

Cheng Yin Chuang

December 17, 2023

Abstract

This study addresses the crucial issue of forest fires, creating extensive damage and posing risk to human lives. In this work, we investigate the Zero-Inflated Model and the Hurdle Model to predict the burned area of forest fires, our analysis includes spatial and temporal factors, Fire Weather Index(FWI) components, and various weather conditions using real-world data collected from the northeast region of Portugal. The Hurdle Negative Binomial model emerged as the most effective, utilizing six inputs Fine Fuel Moisture Code(FFMC), Duff Moisture Code(DMC), Intitial Spread Index(ISI), temperature, relative humidity(RH), and wind speed. This model excels in simulating the areas more likely to be affected by fires, a capability that could significantly enhance the fire department's resource management and response strategies.

1 Introduction

Forest fire has been one of the major environmental concerns, it brings economic and ecological damage. Precise estimation and prediction have become a significant role in controlling such phenomena. Although there are several

ways to achieve such a thing, most of them are expensive and hard to maintain. For example satellite, it costs a lot of money to put a satellite up, plus there is a delay for the satellite to detect a fire and send back the message to the relevant department, the fire might get worse after the process. Therefore, people started to focus on the weather conditions. In this study, the shift towards using weather data has led to exploring models like the Zero-Inflated Model and the Hurdle Model, which show promise in accurately predicting forest fire areas. These models are tested using recent, real-world data from the northeast region of Portugal, incorporating variables such as rain, wind, temperature, and key components of the Fire Weather Index (FWI). The Hurdle Negative Binomial model, in particular, has proven effective. It utilizes inputs like the Fine Fuel Moisture Code (FFMC), Duff Moisture Code (DMC), Initial Spread Index (ISI), temperature, relative humidity (RH), and wind speed. The success of this model in simulating fire-prone areas demonstrates the potential of using weather and environmental data for timely and cost-effective forest fire prediction. This approach could significantly enhance resource management and preparedness strategies for fire departments, potentially reducing the environmental and economic impact of forest fires.

2 Methods and Materials

The following two global metrics will be used to determine how well the model has performed. Mean Absolute Error (MAE) and Mean Mean Absolute Deviation (MAD). MAE measures the average errors in a set of predictions, in easier terms, it is the absolute differences between predicted values($\hat{y}_i$) and the observed values(y_i). A lower MAE value indicates a better fit of the model to the data. The value tells us how big of an error we can expect from the forecast on average. MAD measures how far, on average, each data point is from the mean

of the data, in easier terms, it is the absolute differences between individual data points(x_i) and the mean($\bar{x}$) of all data points. A lower MAD suggests that the data are closer to the mean.

The formula is:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y_i}|$$

and

$$MAD = \frac{1}{N}\sum_{i=1}^{N}|x_i - \bar{x}|$$

Where $\bar{x}$ is the mean of the observed data: $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$

Different data mining models have different purposes, in this study we have considered two data mining models, the zero-inflated negative binomial model and the hurdle model. The Linear Regression Model is easy to interpret and widely used, but the cons are that the model is sensitive to the outlier, and the power of prediction is not as good as other models, which happens in this case. To solve those issues, we can use alternatives, such as Least Absolute Shrinkage and Selection Operator (LASSO), or Random Forest (RF), which is a tree-based model, or nonlinear functions, such as Neural Network (NN). However, RF and NN require much data to train, which is not a recommended method in this case. Therefore, We decided to use the zero-inflated model or Hurdle model, which is a special model that deals with the data with a great amount of zeros.

Zero-Inflated (ZI) model (Lambert, 1992) has been developed to model zero inflation situations when the regular count models such as Poisson or Negative binomial are unrealistic. (Feng, 2021) The choice between the two types of models is often determined by comparing model fit statistics post-fitting both types of models. In ZI model, observations that the outcome is zero have two different origins: "structural" and "sampling". The sampling zeros are from the

usual Poisson or Negative Binomial (NB) distribution, which are assumed that were occurred by chance. The general structure of a ZI model is given as:

$$P(Y_i = y_i) = \begin{cases} \pi_i + (1 - \pi_i)p(y_i = 0; \mu_i), & \text{if } y_i = 0 \\ (1 - \pi_i)p(y_i; \mu_i), & \text{if } y_i > 0 \end{cases}$$

For modeling the count component of a ZI model, Poisson regression assumes the mean equals the variance and NB assumes the variance is greater than the variance under the Poisson model, overdispersion would occur, which does happen in this case. NB regression could be then used to model overdispersed poisson count data, The formula is given by:

$$P(Y_i = y_i) = \begin{cases} \pi_i + (1 - \pi_i) \left[\frac{r}{\mu_i + r}\right]^r & \text{if } y_i = 0, \\ (1 - \pi_i) \frac{\Gamma(y_i + r)}{\Gamma(r) y_i!} \left(\frac{\mu_i}{\mu_i + r}\right)^{y_i} \left(\frac{r}{\mu_i + r}\right)^r & \text{if } y_i > 0 \end{cases}$$

(Feng, 2021)

Where μ_i is the mean of the NB model, π_i is the probability of a structural zero, r is the dispersion parameter, and γ is the gamma function.

The Hurdle model (Mullahy, 1986) can be viewed as a two-component mixture model consisting of a zero mass and the positive observations component following a truncated count distribution, similarly to ZI model, they also have truncated Poisson or truncated NB distribution. The general structure of a hurdle model is given by:

$$P(Y_i = y_i) = \begin{cases} p_i, & \text{if } y_i = 0 \\ (1 - p_i) \frac{p(y_i; \mu_i)}{1 - p(y_i = 0; \mu_i)}, & \text{if } y_i > 0 \end{cases}$$

where p_i is the probability of a subject belonging to the zero component, $p(y_i; \mu_i)$ represents a probability mass function (PMF) for a regular count dis-

tribution with a vector of parameters μ_i. The non-zero count component can follow some distributions to account for overdispersion and NB distribution is the most commonly used. The model is given by:

$$P(Y_i = y_i) = \begin{cases} p_i & y_i = 0, \\ 1 - p_i \frac{\Gamma(y_i+r)}{\Gamma(r)y_i!} \left(\frac{\mu_i}{\mu_i+r}\right)^{y_i} \left(\frac{r}{\mu_i+r}\right)^{r} & y_i > 0, \end{cases}$$

(Feng, 2021)

Monte Carlo simulation is a computational technique that uses random sampling to estimate mathematical functions and mimic the operation of complex systems. The core idea is to estimate the outcomes of a system many times, using random values drawn from probability distributions that represent the uncertainty or variability of the system's input. By looking at the distribution of the outcomes from these simulations, we can make statements about the system's behavior and outcomes(Harrison, 2010). At the base of Monte Carlo simulation are PDFs, functions that define the range of possibilities and the relative probability of those possibilities for a given step in the simulation. A PDF must be a non-negative real-valued function, and its integral over its range must be one. The probability for each distance is given by an exponential distribution, discussed below. The uniform distribution with range from R to S, where $R \leq S$ are real numbers, has a simple PDF: it gives equal probability to every number in its range

$$p(x) = \frac{1}{S - R}, \quad R \leq x \leq S$$

(the interval can also be open or half open as needed)

For sampling purposes we often use the cumulative distribution function(CDF),

Which is defined as an integral of the PDF, for the uniform distribution:

$$P(x) = \int_R^x p(t)\,dt = \frac{x-R}{S-R}, \quad R \le x \le S$$

The CDF tells us the probability that a number sampled randomly from the PDF (Harrison, 2010).

3 Results

Before analyzing the data, we need to convert nominal variables such as month and day into integers. For variable area, 247 observations are zero. In this case, zero does not mean there is no fire happened, it means that fire occurrence with an area lower than $1ha/100 = 100m^2$ was burned. (Cortez & Morais, 2007) Since we are doing the ZIM and HM, we do not need to transform the skewed issue. However, the response variable need to be count number, which can not be a continuous number, then we decide to round the number up because I do not think it will make any huge difference.

3.1 Model comparison

We separate the 517 observations into 361 observations for the training set and 156 observations for the validation set. First, we train the data with the training data set and use the testing data set to calculate how well the prediction is and use two metrics to see how well the prediction of different models performance. Because the variance is greater than the mean, we choose to compare the Zero-Inflated Negative Binomial(ZINB) model and the Hurdle Negative Binomial(HNB) model. Since the response variable needs to be count data, which can not have anything after the decimal, I decided to round the numbers because I did not think the result would change a lot before and after.

```
Count model coefficients (poisson with log link):
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.1152931  0.8449255  -7.238 4.56e-13 ***
X            0.1680060  0.0076128  22.069  < 2e-16 ***
Y           -0.0761805  0.0140181  -5.434 5.50e-08 ***
month        0.3468055  0.0168799  20.545  < 2e-16 ***
day          0.1676005  0.0082338  20.355  < 2e-16 ***
FFMC         0.0775854  0.0094607   8.201 2.39e-16 ***
DMC          0.0070161  0.0003697  18.979  < 2e-16 ***
DC          -0.0027303  0.0001782 -15.321  < 2e-16 ***
ISI         -0.0594745  0.0066032  -9.007  < 2e-16 ***
temp         0.0388473  0.0042675   9.103  < 2e-16 ***
RH          -0.0325124  0.0015430 -21.072  < 2e-16 ***
wind         0.0298294  0.0103863   2.872  0.00408 **
rain        -1.3261110  0.5151105  -2.574  0.01004 *

Zero-inflation model coefficients (binomial with logit link):
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.4117671  4.7553336  -0.507   0.6120
X           -0.0256426  0.0556096  -0.461   0.6447
Y           -0.0482072  0.1065382  -0.452   0.6509
month       -0.0783853  0.1182994  -0.663   0.5076
day         -0.0393764  0.0535141  -0.736   0.4618
FFMC         0.0501953  0.0543347   0.924   0.3556
DMC         -0.0003483  0.0026734  -0.130   0.8963
DC           0.0004186  0.0013653   0.307   0.7591
ISI         -0.0040083  0.0314908  -0.127   0.8987
temp        -0.0310929  0.0314499  -0.989   0.3228
RH          -0.0015396  0.0097343  -0.158   0.8743
wind        -0.1263962  0.0670625  -1.885   0.0595 .
rain        -1.0201260  2.1143559  -0.482   0.6295
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 1: ZINB model result

The table in figure 1 is the result of ZINB and we can see that all the variables were classified as significant in the count model part if we chose the variables that were lower than 0.05. For the binomial part below that determines whether the variables are statistically significant enough to tell if a variable belongs to structural zero or sample zero, and no variables were statistically significant enough to tell the difference.

The table in figure 2 is the result of HNB and both parts performed very similarly to the ZINB. However, there is one notable point in the binomial part. The wind's p-value is lower than 0.05, meaning it is significant enough to tell the difference between an observation in structural zero or sampling zero. Moreover, we test the MAE and MAD of the two methods. The result is below.

We can see the MAE and MAD of HNB and ZINB in Figure 3, based on the result and the difficulty of interpreting of the table, choosing the Hurdle model is a better choice in this case. Before moving on to the simulation, we

```
Count model coefficients (truncated poisson with log link):
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.1591822  0.8666028  -7.107 1.18e-12 ***
X            0.1683099  0.0076351  22.044  < 2e-16 ***
Y           -0.0762455  0.0140340  -5.433 5.54e-08 ***
month        0.3484148  0.0169466  20.560  < 2e-16 ***
day          0.1679981  0.0082353  20.400  < 2e-16 ***
FFMC         0.0778674  0.0097079   8.021 1.05e-15 ***
DMC          0.0070202  0.0003699  18.978  < 2e-16 ***
DC          -0.0027381  0.0001783 -15.353  < 2e-16 ***
ISI         -0.0594518  0.0066379  -8.956  < 2e-16 ***
temp         0.0390550  0.0042767   9.132  < 2e-16 ***
RH          -0.0324781  0.0015424 -21.057  < 2e-16 ***
wind         0.0297267  0.0103998   2.858  0.00426 **
rain        -1.4696739  0.6578003  -2.234  0.02547 *
Zero hurdle model coefficients (binomial with logit link):
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.4233588  3.2393489  -0.439   0.6604
X            0.0333129  0.0550280   0.605   0.5449
Y            0.0314153  0.1042976   0.301   0.7633
month        0.0947083  0.1152949   0.821   0.4114
day          0.0388131  0.0525928   0.738   0.4605
FFMC        -0.0048194  0.0345801  -0.139   0.8892
DMC          0.0003756  0.0026215   0.143   0.8861
DC          -0.0005685  0.0013284  -0.428   0.6687
ISI         -0.0095384  0.0289900  -0.329   0.7421
temp         0.0253505  0.0306127   0.828   0.4076
RH          -0.0011855  0.0094447  -0.126   0.9001
wind         0.1331939  0.0655889   2.031   0.0423 *
rain         0.9420355  1.9661785   0.479   0.6319
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 2: HNB model result

mae_zim <dbl>	mae_hurdle <dbl>	mad_zim <dbl>	mad_hurdle <dbl>
25.9363	25.92068	10.61394	10.59393

Figure 3: MAE and MAD comparison

have to choose the variables that we want to fit into the model, we exclude the X, Y, month, and day because X and Y could vary from place to place, and the weather of month and day could also vary from country to country. We refit the model again to see what variables were significant enough for the model, the final variables that we chose were FFMC, DMC, ISI, temp, RH, and wind, and for the binomial part of the model we only chose wind as the only variable.

3.2 Simulation

We set all the variables as normal distribution randomly producing 1000 observations and reproducing if the values are less than the minimum of the original

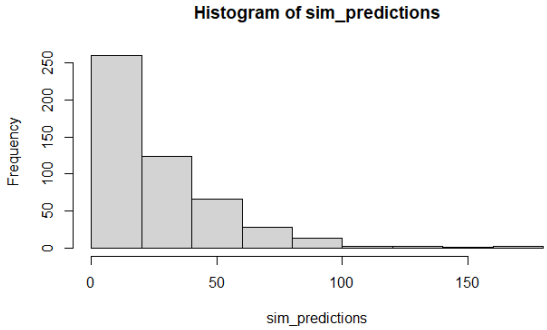

Figure 4: Simulation histogram

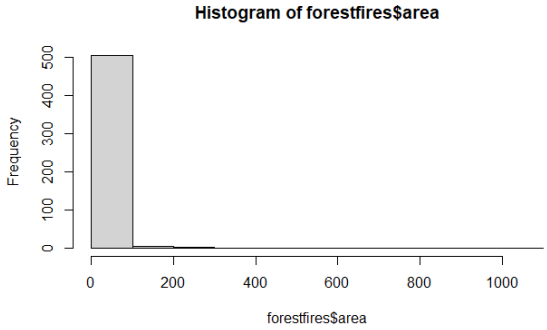

Figure 5: Original histogram

values, the following is the histogram of the predicted simulation area.

We can see the plot in figure 4 is highly skewed to the right, which is similar to the original case(figure 5). And that is what we want to see.

4 Discussion and Summary

Forest fires cause significant environmental damage, not only threatening the animals living in the forest but also to human beings. People starting to invent different techniques to predict forest fires. In this work, we use real-world data, from the northeast region of Portugal, which was used in the experiments to

predict future scenarios. This dataset includes different weather conditions and some components from the Canadian Fire Weather Index. We use the Zero-Inflated Model and the Hurdle Model those invent to deal with excess amounts of zeros. The proposed solution is based on the Hurdle Model, which is slightly better than the Zero-Inflated Model in this case. Most variables used for prediction, such as weather conditions and FWI components, were significantly correlated to the area burned While the simulation results were encouraging, they also revealed certain limitations. The dataset, particularly skewed towards August and September, lacked sufficient data points for other months like January, May, and November, restricting the model's comprehensiveness. Additionally, the assumption of a normal distribution for all variables in the simulation might not accurately reflect the true nature of each variable. Future improvements could involve tailoring the distribution types for different variables to enhance the precision of the predictions. Moreover, having feedback from the firefighting managers will also improve the model's prediction effectively. Lastly, although the simulations didn't address the outlier, studying the patterns of the outlier is still necessary.

5 References

1. Cortez, P., & Morais, A.D. (2007). A data mining approach to predict forest fires using meteorological data.

2. Faraway, J. J. (2005). Extending the linear model with R generalized linear, mixed effects, and nonparametric regression models. Chapman & Hall.

3. Feng, C.X. (2021) A comparison of zero-inflated and hurdle models for modeling zero-inflated count data. J Stat Distrib App 8, 8. https://doi.org/10.1186/s40488-021-00121-4

4. Harrison, R.L. (2010). Introduction To Monte Carlo Simulation. AIP

Conference Proceedings, 1204, 17-21. doi: 10.1063/1.3295638

5. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2022). An introduction to statistical learning: With applications in R. Springer.

6. Lambert, D. (1992). Zero-inflated Poisson regression, with an application to defects in manufacturing. Technometrics, 34(1), 1. https://doi.org/10.2307/1269547

7. Mullahy, J. (1986): Specification and testing of some modified count data models. J. Econ. 33, 341–365.

8. S. Taylor and M. Alexander (2006). Science, technology, and human factors in fire danger rating: the Canadian experience. International Journal of Wildland Fire, 15:121–135.

6 Appendix

Code

```
# Import library
library(readxl)
library(dplyr)
library(ggplot2)
library(cowplot)
# Random forest library
library(randomForest)
library(psych)
library(keras)
# Lasso
library(glmnet)
library(caret)
library(boot)
library(car)
library(scales)
```

```
library(glm2)
# Neural Network
library(nnet)
library(tidyr)

```

```{r}
#import data
forestfires <- read.csv("C:\\Users\\54088\\OneDrive\\
    \\SW\\S5 2023\\STAT 4893W\\Second Project\\forestfires.
    csv")

# Explaination of the covariates
forestfires
#   1. X - x-axis spatial coordinate within the Montesinho
    park map: 1 to 9
#   2. Y - y-axis spatial coordinate within the Montesinho
    park map: 2 to 9

#   3. month - month of the year: 'jan' to 'dec'
#   4. day - day of the week: 'mon' to 'sun'

#   5. FFMC - Fine Fuel Moisture Code index from the FWI
    system: 18.7 to 96.20
#   6. DMC - Duff Moisture index from the FWI system: 1.1 to
     291.3
#   7. DC - Drought Code index from the FWI system: 7.9 to
    860.6

```

```
#   8. ISI - Initial Spread Index from the FWI system: 0.0
    to 56.10

#   9. temp - Temperature in Celsius degrees: 2.2 to 33.30
#   10. RH - Relative Humidity in %: 15.0 to 100
#   11. wind - Wind Speed in km/h: 0.40 to 9.40
#   12. rain - outside rain in mm/m2 : 0.0 to 6.4
#   13. area - the burned area of the forest (in ha): 0.00
    to 1090.84
#   (this output variable is very skewed towards 0.0, thus
    it may make
#   sense to model with the logarithm transform).

# Rearrange the variables
forestfires <- forestfires %>%
  mutate(month = case_when(month %in% "jan" ~ 1,
                           month %in% "feb" ~ 2,
                           month %in% "mar" ~ 3,
                           month %in% "apr" ~ 4,
                           month %in% "may" ~ 5,
                           month %in% "jun" ~ 6,
                           month %in% "jul" ~ 7,
                           month %in% "aug" ~ 8,
                           month %in% "sep" ~ 9,
                           month %in% "oct" ~ 10,
                           month %in% "nov" ~ 11,
                           month %in% "dec" ~ 12))

forestfires <- forestfires %>%
  mutate(day = case_when(day %in% "mon" ~ 1,
```

```
                            day %in% "tue" ~ 2,
                            day %in% "wed" ~ 3,
                            day %in% "thu" ~ 4,
                            day %in% "fri" ~ 5,
                            day %in% "sat" ~ 6,
                            day %in% "sun" ~ 7))

forestfires$RH <- as.numeric(forestfires$RH)

#forestfires$X <- as.factor(forestfires$X)
#forestfires$Y <- as.factor(forestfires$Y)
#forestfires$month <- as.factor(forestfires$month)
#forestfires$day <- as.factor(forestfires$day)
```

```{r}
EDA_forest <- forestfires
par(mfrow = c(1, 9))

# X
ggplot(EDA_forest, aes(x = X)) +
  geom_histogram(binwidth = 1, fill = "lightblue", color = "
    black") +
  theme_minimal() +
  ggtitle("Distribution of X") +

# Y
ggplot(EDA_forest, aes(x = Y)) +
  geom_histogram(binwidth = 1, fill = "lightblue", color = "
    black") +
```

```
  theme_minimal() +
  ggtitle("Distribution of Y")

# FFMC
ggplot(EDA_forest, aes(x = FFMC)) +
  geom_histogram(binwidth = 2, fill = "lightblue", color = "
    black") +
  theme_minimal() +
  ggtitle("Distribution of FFMC")

# DMC
ggplot(EDA_forest, aes(x = DMC)) +
  geom_histogram(binwidth = 10, fill = "lightblue", color =
    "black") +
  theme_minimal() +
  ggtitle("Distribution of DMC")

# DC
ggplot(EDA_forest, aes(x = DC)) +
  geom_histogram(binwidth = 25, fill = "lightblue", color =
    "black") +
  theme_minimal() +
  ggtitle("Distribution of DC")

# ISI
ggplot(EDA_forest, aes(x = ISI)) +
  geom_histogram(binwidth = 2, fill = "lightblue", color = "
    black") +
  theme_minimal() +
  ggtitle("Distribution of ISI")
```

```r

# temp
ggplot(EDA_forest, aes(x = temp)) +
  geom_histogram(binwidth = 1, fill = "lightblue", color = "
    black") +
  theme_minimal() +
  ggtitle("Distribution of temp")

# RH
ggplot(EDA_forest, aes(x = RH)) +
  geom_histogram(binwidth = 1, fill = "lightblue", color = "
    black") +
  theme_minimal() +
  ggtitle("Distribution of RH")

# wind
ggplot(EDA_forest, aes(x = wind)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue", color =
     "black") +
  theme_minimal() +
  ggtitle("Distribution of wind")

# rain
ggplot(EDA_forest, aes(x = rain)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue", color =
     "black") +
  theme_minimal() +
  ggtitle("Distribution of rain")

# area
```

```
ggplot(EDA_forest, aes(x = area)) +
  geom_histogram(binwidth = 100, fill = "lightblue", color =
     "black") +
  theme_minimal() +
  ggtitle("Distribution of area")

```

```{r}

# EDA for month
month1 <- sum(forestfires$month == 1)
month2 <- sum(forestfires$month == 2)
month3 <- sum(forestfires$month == 3)
month4 <- sum(forestfires$month == 4)
month5 <- sum(forestfires$month == 5)
month6 <- sum(forestfires$month == 6)
month7 <- sum(forestfires$month == 7)
month8 <- sum(forestfires$month == 8)
month9 <- sum(forestfires$month == 9)
month10 <- sum(forestfires$month == 10)
month11 <- sum(forestfires$month == 11)
month12 <- sum(forestfires$month == 12)

colors <- c("#F8766D", "#DE8C00", "#B79F00", "#7CAE00", "#00
    BA38", "#00C08B", "#00BFC4","#00B4F0", "#619CFF", "#
    C77CFF", "#F564E3", "#FF64B0")
barplot(table(forestfires$month), col = colors, main = "
    Countplot for the days in the month", names.arg = c("jan"
```

```
      , "feb", "mar", "apr", "may", "jun", "jul", "aug", "sep",
       "oct", "nov", "dec"))
colors <- c("#F8766D", "#C49A00", "#53B400", "#00C094", "#00
      B6EB", "#A58AFF", "#FB61D7")
barplot(table(forestfires$day), col = colors, main = "Count
      plot for the days in the week", names.arg = c("mon", "tue
      ", "wed", "thu", "fri", "sat", "sun"))


# Test the Multicollinearity

cor(forestfires[, 1:13])
model0 <- lm(area ~ FFMC + DMC + DC + ISI + temp + RH + wind
       + rain, data = forestfires)

vif(model0)

model1 <- lm(area ~ FFMC + DMC + ISI + temp + RH + wind +
      rain, data = forestfires)

vif(model1)
# We can see that DC and month shows the strong linear
      dependence

```

```{r}
# Zero-inflated poisson model
zim_fires <- forestfires
zim_fires$area <- round(zim_fires$area)
```

```
194
195 library(pscl)
196 set.seed(4893)
197 nr = nrow(zim_fires)
198 train_indices = sample(nr, nr*0.7)
199 train_zim <- zim_fires[train_indices, ]
200 test_zim <- zim_fires[-train_indices, ]
201
202 # The former part specifies the count model formula implies
      the predictive variable affect the count model and the
      latter part after | is the zero-inflated model, implies
      what predictive variable affect the probability of
      getting a zero.
203 model_zim <- zeroinfl(area ~ .|., data = train_zim, family =
       "negbin")
204 summary(model_zim)
205
206 # Large pearson residual indicates poor fit or outliers.
207 # For the non-zero counts of the area variable. Every
      predictive variables have strong association with the
      area. But, when it comes to the probability of the zeros,
       the increase of variables "day", "DC", "ISI", the
      probability of area being zero increase.
208
209 #(poisson with log link): meaning the count part is modeled
      using a Poisson distribution. log link helps with making
      sure that the result is positive integer.
210
211 #(binomial with logit link): Binomial (0 or 1). The logit
      translates the linear combination of predictors into a
```

```
      probability between 0 and 1.
212
213 # The log likelihood shows how good the model fits the data,
       the higher the number is, the greater the model fits the
       data.
214
215 predict_zim <- predict(model_zim, newdata = test_zim, type =
       "count")
216
217 #MAE
218 mae_zim <- mean(abs(predict_zim - test_zim$area))
219 mae_zim
220
221
222 #MAD
223 mad_zim <- median(abs(predict_zim - mean(test_zim$area)))
224 mad_zim
225
226
227
228 ```
229
230 ```{r}
231 # hurdle model
232 hurdle_fires <- forestfires
233 hurdle_fires$area <- round(hurdle_fires$area)
234
235 set.seed(4893)
236 nr = nrow(hurdle_fires)
237 train_indices = sample(nr, nr*0.7)
```

```r
train_hurdle <- hurdle_fires[train_indices, ]
test_hurdle <- hurdle_fires[-train_indices, ]

model_hurdle <- hurdle(area ~ .|., data = train_hurdle,
    family = "negbin")
summary(model_hurdle)

predict_hurdle <- predict(model_hurdle, newdata = test_
    hurdle, type = "count")
#MAE
mae_hurdle <- mean(abs(predict_hurdle - test_zim$area))

#MAD
mad_hurdle <- median(abs(predict_hurdle - mean(test_hurdle$
    area)))
# Compared
data.frame(
  cbind(mae_zim, mae_hurdle),
  cbind(mad_zim, mad_hurdle)
)
mae_zim

```

```{r}
# Monte Carlo simulation
# We want to do variables selection, for count model, since
    we need the variables be general, we exclude the X, Y,
    month, day because it can vary from place to place. We
    rank in RH, DMC, DC, temp, FFMC, ISI, wind and rain in
```

```
      order.
262 # For zero-hurdle model, wind is the only model that help us
      .
263
264 # select the top three important variables
265 hurdle_model <- hurdle(area ~ FFMC + DMC + DC + ISI + temp +
       RH + wind + rain|wind, data = train_hurdle, family = "
      negbin")
266 summary(hurdle_model)
267 # Remove some variables p-value larger than 2^e-10 <- DC and
       rain
268
269 # simulation preprocess
270 set.seed(4052)
271 sim_FFMC <- rnorm(n = 1000, mean = mean(hurdle_fires$FFMC),
      sd = sd(hurdle_fires$FFMC))
272 sim_DMC <- rnorm(n = 1000, mean = mean(hurdle_fires$DMC), sd
       = sd(hurdle_fires$DMC))
273 sim_ISI <- rnorm(n = 1000, mean = mean(hurdle_fires$ISI), sd
       = sd(hurdle_fires$ISI))
274 sim_temp <- rnorm(n = 1000, mean = mean(hurdle_fires$temp),
      sd = sd(hurdle_fires$temp))
275 sim_RH <- rnorm(n = 1000, mean = mean(hurdle_fires$RH), sd =
       sd(hurdle_fires$RH))
276 sim_wind <- rnorm(n = 1000, mean = mean(hurdle_fires$wind),
      sd = sd(hurdle_fires$wind))
277
278
279 while(any(sim_FFMC < 18)){
280   sim_FFMC[sim_FFMC < 18] <- rnorm(sum(sim_FFMC < 18), mean
```

```
      = mean(hurdle_fires$FFMC), sd = sd(hurdle_fires$FFMC))
}
while(any(sim_DMC < 1)){
  sim_DMC[sim_DMC < 1] <- rnorm(sum(sim_DMC < 1), mean =
    mean(hurdle_fires$DMC), sd = sd(hurdle_fires$DMC))
}
while(any(sim_ISI < 0)){
  sim_ISI[sim_ISI < 0] <- rnorm(sum(sim_ISI < 0), mean =
    mean(hurdle_fires$ISI), sd = sd(hurdle_fires$ISI))
}
while(any(sim_temp < 2)){
  sim_temp[sim_temp < 2] <- rnorm(sum(sim_temp < 2), mean =
    mean(hurdle_fires$temp), sd = sd(hurdle_fires$temp))
}
while(any(sim_RH < 15)){
  sim_RH[sim_RH < 15] <- rnorm(sum(sim_RH < 15), mean = mean
    (hurdle_fires$RH), sd = sd(hurdle_fires$RH))
}
while(any(sim_wind < 0)){
  sim_wind[sim_wind < 0] <- rnorm(sum(sim_wind < 0), mean =
    mean(hurdle_fires$wind), sd = sd(hurdle_fires$wind))
}

sim_wind <- round(sim_wind, digits = 1)
sim_RH <- round(sim_RH, digits = 0)
sim_FFMC <- round(sim_FFMC, digits = 1)

sim_data <- data.frame(FFMC = sim_FFMC, DMC = sim_DMC, ISI =
     sim_ISI, temp = sim_temp, RH = sim_RH, wind = sim_wind)

```

```
304
305 # Simulation
306 # training and testing set
307 model_hurdle <- hurdle(area ~ FFMC + DMC + ISI + temp + RH +
        wind|wind, data = train_hurdle, family = "negbin")
308 summary(model_hurdle)
309
310 predict_hurdle <- predict(model_hurdle, newdata = test_
      hurdle, type = "count")
311
312 # simulation set
313 hurdle_model <- hurdle(area ~ FFMC + DMC + ISI + temp + RH +
        wind|wind, data = train_hurdle, family = "negbin")
314 summary(hurdle_model)
315 sim_predictions <- predict(hurdle_model, newdata = sim_data,
        type = "count")
316
317 # Compare.
318 hist(predict_hurdle)
319 hist(sim_predictions)
320 hist(forestfires$area)
```

Additional graph

```
Call:
hurdle(formula = area ~ FFMC + DMC + DC + ISI + temp + RH + wind +
    rain | wind, data = train_hurdle, family = "negbin")

Pearson residuals:
    Min      1Q  Median      3Q     Max
-1.1741 -0.9057 -0.8086 -0.3638 36.8208

Count model coefficients (truncated poisson with log link):
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.523e+00  6.875e-01  -2.216  0.02669 *
FFMC         5.765e-02  7.734e-03   7.454 9.03e-14 ***
DMC          5.461e-03  3.231e-04  16.902  < 2e-16 ***
DC           1.282e-04  9.519e-05   1.346  0.17822
ISI         -7.402e-02  5.911e-03 -12.521  < 2e-16 ***
temp         2.646e-02  3.871e-03   6.836 8.14e-12 ***
RH          -3.574e-02  1.470e-03 -24.314  < 2e-16 ***
wind         1.055e-01  9.317e-03  11.326  < 2e-16 ***
rain        -1.830e+00  6.573e-01  -2.784  0.00537 **
Zero hurdle model coefficients (binomial with logit link):
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.5799     0.2586  -2.242   0.0249 *
wind          0.1203     0.0598   2.012   0.0442 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Number of iterations in BFGS optimization: 17
```

Figure 6: variables selection table

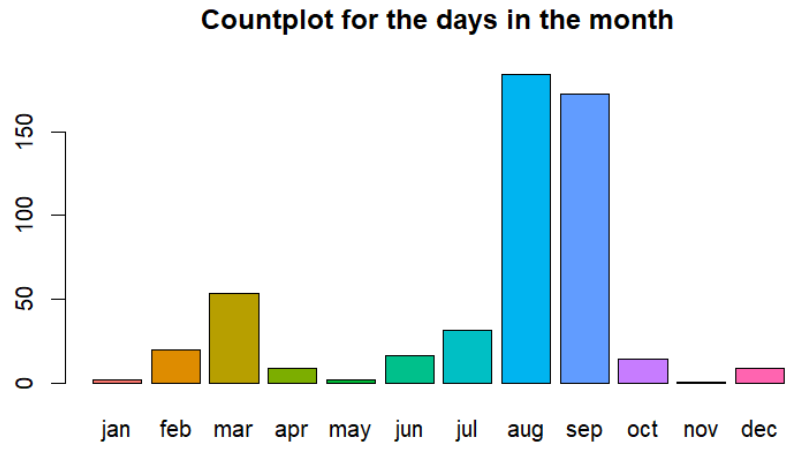

Figure 7: Countplot for month

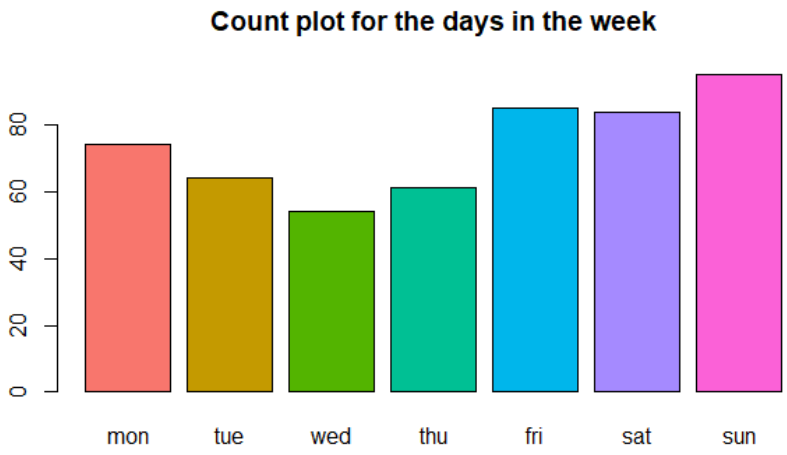

Figure 8: Countplot for day