Fundamentals of Data Science: Prediction, Inference, Causality

MS&E 226

Predict the size of wildfires

Author:
Thibaut Badoual
Melvin Juwono

Fall 2020

1 Investigating and exploring your data

1.1 Introduction

The selected dataset originated from the Forest Service Research Data Archive and contains information regarding **fires across the United States**. The original dataset contained data from 1.88 million fires, from which 55,367 were randomly selected and enhanced to form the selected dataset. Each data entry was combined with historical weather data, historical vegetation data, and a "remoteness" metric giving a sense of how close the fire was from a city. Each data point consists of 43 variables.

There is a mix of data collected digitally and manually; digitally collected data such as meteorological data are expected to be reliable, whereas manually collected data such as cause of fire, putout time, and discovery date are potentially unreliable and may be incomplete.

1.2 Pre-processing

For pre-processing, we removed columns not useful for our data analysis including ID, fire name, various alternative date formats, and weather station related metrics. These variables were either uncorrelated to fire size or could be derived from other variables in the dataset. After some analysis, the year column was also dropped; as shown in Figure 1 of the appendix, the month plays a more important role than the year, although this assumption may be questionable in the coming years with global warming.

Data that would not have been available at the start of the fire such as putout time and other information collected on the fire's containment date, were also dropped as these would not be available in practice when predicting fire size. And, duplicate rows and rows with missing data (meteorological and vegetation) were removed - we did consider several methods to fill the missing data (local regression to predict the missing value, etc).

Lastly, we selected to focus only on the ten states that had the most recorded wildfires as some states were significantly underrepresented. The resulting dataset consists of **18,202 data entries and 20 variables**. Note that we also add a binary response variable based on fire size class (as discussed below), to bring the total number of columns to 21.

1.3 Choice of the continuous and binary response variable

The continuous response variable we plan to use for the prediction task is fire size. Fire size is defined as the number of acres within the perimeter of the fire. This variable is useful to predict as models can then be used to ensure the appropriate amount of preparation and reduce the time it takes to contain the fire (hence limiting damage).

The binary response variable we plan to use for the prediction task is a variation of fire size class. Fire size class categorizes the fire size, with the following translation: A) greater than 0 but less than or equal to 0.25 acres, B) 0.26-9.9 acres, C) 10.0-99.9 acres, D) 100-299 acres, E) 300 to 999 acres, F) 1000 to 4999 acres, and G) 5000+ acres. Different response plans can be created and quickly deployed based on fire size class without a need to know the expected fire size. We chose to set fire size classes **above C to 1, and C or below to 0**.

1.4 Estimated weight of each covariate

At first glance, variables we believed should have a significant impact on the response variables include cause of fire, remoteness, month of discovery, vegetation, and meteorological data.

We hypothesized that cause of fire should have an impact as there is an element of planned vs. accidents, and expected vs. unexpected that affects the starting magnitude of the fire and the response, if any. This is supported by Figure 2 in the appendix as some causes, such as lightning, have a high probability of causing very large fires (local storm, difficulties of intervention, strong winds, etc.). We also hypothesized that remoteness will have a significant impact as fires closer to a city are likely to have a quicker response time and hence contain the fire before it reaches maximum size. This was not supported by our findings.

Amongst the other variables mentioned, the most noteworthy correlation with fire size appears to be humidity as shown in Figure 3 of the appendix. Intuitively, lower humidity leads to drier fuels, increasing the risk of fire. Finally, it is important to note that weather data is often linked to each other and to their geographic coordinates, which is quite predictable and expected.

1.5 Challenges raised

Predominantly, this dataset is interesting as we can use it to predict fire size or fire size class of a fire that was just discovered. In addition to prediction, the following are some other questions we might be able to answer using this dataset:

- Which types of vegetation are more susceptible to fires? Or are correlated with larger fire sizes?
- Does the impact of vegetation on fire size differ with changes in meteorological data?
- Can we predict how dangerous a fire is to society by considering fire size and remoteness?
- Are there specific causes of fires that consistently lead to higher fire sizes?
- Using some columns we had previously dropped for the prediction task (e.g. putout time), are there specific states better prepared to contain wildfires?
- Can we evaluate the response capacity of the different American states according to the number of very large fires in their regions?

As highlighted by the questions listed above, this dataset is exciting because of its potential to analyze wildfire response strategies used by the state the fire originated in. Furthermore, with a better understanding of how each variable contributes to fire size, government agencies can assess risk better and plan accordingly.

To add to the discussion, one variable that would be good to add to the dataset is whether resources were deployed to fight the fire. If we knew whether the fire was contained naturally or with the help of fire departments, then an interaction that may be worth looking into is remoteness because the closer a fire is to a city, the shorter the reaction time.

2 Prediction

2.1 Acceptable performances

A good outcome for our predictive models would be high accuracy when predicting the continuous response variable (fire size), and high recall / sensitivity when predicting the binary response variable (fire size class).

Higher accuracy is desired when predicting fire size so that the potential amount of damage can be estimated. However, a precise number is not needed as it is enough to get an idea of how big the fire size is. For the binary response variable (fire size class), high sensitivity is desired as fires greater than 100 acres (class D or above) may pose a significantly larger threat and so response teams will want to be prepared. Response teams may prefer to be over-prepared than under-prepared and therefore the model should be willing to trade off an increase in the number of false positives for a decrease in the number of false negatives.

2.2 Evaluation strategy

Our baseline model for the regression task is a linear model (OLS) involving all covariates without any transformations or interactions. For the classification task, we used logistic regression also without transformations or interactions. We believe both models can be improved upon significantly as we suspect both will have high bias due to the absence of relevant interactions and effective transformations.

Building on the baseline models, we tried different transformations, further data manipulation, various interactions, and a combination of all. When testing each modified model, we used 10-fold cross-validation and assessed the Root Mean Squared Error (RMSE), accepting the modifications that led to lower RMSE values.

The first model tested for the regression task takes into account all possible interactions between the covariates. Barring surprises, this model should not generate convincing results. By taking into account an abundance of interactions, the model may gradually stop reflecting the general trend of the variables but rather their noise. We therefore expect to obtain a model that overfits the training data - having high variance.

Models two and three then took into account all transformations and all relevant interactions determined respectively. It is hoped that a clear improvement will be observed when compared to the baseline model (lower bias and lower variance). Finally, the final model takes all improvements into account.

2.3 Transformations performed

When considering the different transformations performed, we initially thought about the interactions that can exist between the different covariates. If one modifies the behavior of the other with the size or class of fire, an interaction term must be added.

We first took note that, for meteorological data, creating an interaction term between data taken over different time periods makes no physical sense. If it is necessary to create an interaction between wind and humidity, for example, then this should be done on data taken the same number of days before the fire starts. With this in mind, we believed that the likelihood of having a fire in dry, windy weather will be greater than the likelihood of having a fire in wet, windy weather (and vice versa). Similarly, a fire has a greater chance of spreading in hot, dry weather than in hot, humid weather. Hence, we included interaction terms between humidity and wind, as well as humidity and temperature.

We also noted that the relationship between precipitation and fire size does not depend on other meteorological data. If it rains, the probability of having a large fire will vary little, or not at all, with temperature, humidity, or wind. Hence, no interaction terms related to precipitation were included.

Finally, vegetation plays an important role in the relationship between weather data and fire size. Even in very dry, windy, rainless weather, a fire has little chance of spreading in a desert. However, adding the interaction between vegetation and temperature seems to decrease the final accuracy. Therefore, we will not retain the latter.

Another factor we considered revolved around latitude and longitude, which are completely independent of one another. Longitude appears to be the more important variable as it represents the distance to the equator and is therefore more indicative of the local climate.

More mathematical transformations were also carried out. The evolution of the size of the fire as a function of each data led us to take the logarithm of the temperature (converted into Kelvin) and precipitation (to which we have added 1 to overcome the difficulty of log(0)). Then, in view of the previous remarks, we decided not to take into account the latitude. Finally, it seemed more relevant to work with seasons instead of months. Indeed, the seasons provide general and annual information on weather data, whereas the time scale of the month is more sensitive to exceptional events that can distort the general trend.

Table 1: Results of the different models on the train dataset

Model	RMSE	Rsquared
Baseline	7913.881	0.03418
Model 1	1.7*e12	0.02842
Model 2	7639.844	0.0310
Model 3	7991.487	0.03908
Final model	7898.117	0.03116

In addition to OLS models, we experimented with Lasso and Ridge Regression. We expected the resulting predictions to decrease variance as these models included a regularization term and normalized numerical covariates. We tested a sequence of values for the regularization term and was able to determine the optimal value.

Table 2: Ridge and Lasso regression

Regression	lambda	RMSE
Lasso	39.810	6902.1
Ridge	2511.8	6903.7

As far as classification is concerned, we tested two different models: logistic regression and random forest. For each of them, we plotted the receiver operating characteristic curve (R0C) (presented in the appendix). This curve deserves a quick reflection. First of all, the logistic regression model seems to have a great sensitivity as well as a high precision. On the other hand, the random forest model appears to have almost perfect sensitivity and precision. The ideal model? Not really, it rather reveals a very marked overfitting. The noise seems to be more modeled than the overall trend.

Table 3: Performances of each model for the classification task

Classification model	random forest	logistic regression
Accuracy %	0.9989	0.9283
Area under curve	0.9999	0.8308
Sensitivity	0.9997	0.7617

3 Conclusion

To conclude, the model determined to be the best suited for the regression task (predicting fire size) was Lasso Regression, which had a RMSE of 6902.1, marginally better than Ridge Regression at 6903.7 and a significant improvement from the baseline model, 7913.9. The RMSE values for the models tested for the regression task are shown in Tables 1 and 2 in the appendix. We think that predictions using the best model on a test set should result in a similar RMSE because the regularization term should decrease variance but not bias.

For the classification task (predicting fire size class), the model determined to be best suited was random forest, with a sensitivity of 0.9997 (Table 3). This was an improvement from logistic regression, which had a sensitivity of 0.7617. We think that predictions using the best model on a test set should result in a lower sensitivity as we believe the model may have overfit the training data leading to high variance.

4 Appendix

```
$ fire_size         <dbl> 10.00, 3.00, 60.00, 1.00, 1.00, 1.00, 8.30, 1.00, 1.00, 1.00, 5.00, 40.00, 5.00, 1.…
$ fire_size_class   <chr> "C", "B", "C", "B", "B", "B", "B", "B", "B", "B", "B", "C", "B", "B", "B", "C", "B"…
$ stat_cause_descr  <chr> "Missing/Undefined", "Arson", "Arson", "Campfire", "Arson", "Miscellaneous", "Debri…
$ latitude          <dbl> 18.10507, 35.03833, 34.94780, 30.90472, 35.90031, 48.83940, 30.84534, 31.76738, 33.…
$ longitude         <dbl> -66.75304, -87.61000, -88.72250, -93.55750, -92.06118, -99.71850, -83.12799, -93.14…
$ state             <chr> "PR", "TN", "MS", "TX", "AR", "ND", "GA", "LA", "NM", "NC", "TX", "MS", "PR", "SC",…
$ discovery_month   <chr> "Feb", "Dec", "Feb", "Nov", "Aug", "Apr", "Mar", "Jul", "Feb", "Feb", "Dec", "Apr",…
$ Vegetation        <chr> "Open Shrubland", "Polar Desert/Rock/Ice", "Secondary Tropical Evergreen Broadleaf …
$ Temp_pre_30       <dbl> 24.480974, 7.553433, 4.971930, 16.851939, 26.655241, 4.600950, 8.410983, 26.384493,…
$ Temp_pre_15       <dbl> 24.7169231, 7.0100000, 5.7827660, 16.9977827, 27.2648699, 6.8618785, 9.0071926, 26.…
$ Temp_pre_7        <dbl> 24.9025974, 0.3435294, 5.5587500, 20.4347826, 28.9680639, 6.0533333, 8.2090000, 26.…
$ Wind_pre_30       <dbl> 4.341807, 2.709764, 3.364499, 1.331257, 1.768074, 6.380760, 1.988671, 1.771639, 6.2…
$ Wind_pre_15       <dbl> 3.492857, 2.881707, 2.923830, 1.472949, 1.705297, 6.334254, 1.745012, 1.629743, 6.0…
$ Wind_pre_7        <dbl> 3.262092, 1.976471, 2.695833, 1.424783, 1.827944, 6.645333, 2.224500, 2.019048, 5.7…
$ Hum_pre_30        <dbl> 78.21659, 70.84000, 75.53163, 72.89948, 68.31902, 64.60651, 71.26087, 79.94001, 38.…
$ Hum_pre_15        <dbl> 76.79375, 65.85891, 75.86861, 75.06138, 67.57542, 55.94304, 69.28103, 79.94212, 45.…
$ Hum_pre_7         <dbl> 76.38158, 55.50588, 76.81283, 77.92462, 65.07784, 54.33784, 64.79798, 67.63095, 40.…
$ Prec_pre_30       <dbl> 0.0, 59.8, 168.8, 28.4, 6.6, 12.3, 76.3, 62.8, 10.0, 49.0, 0.0, 0.0, 0.0, 133.0, 52…
$ Prec_pre_15       <dbl> 0.0, 8.4, 42.2, 27.5, 3.3, 1.8, 26.2, 4.3, 8.0, 39.9, 0.0, 0.0, 0.0, 0.0, 38.4, 85.…
$ Prec_pre_7        <dbl> 0.0, 0.0, 18.1, 1.2, 0.0, 0.0, 8.4, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 2.3, 41.4, 9…
$ binary_class      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,…
```

Figure 1: The wildfire dataset

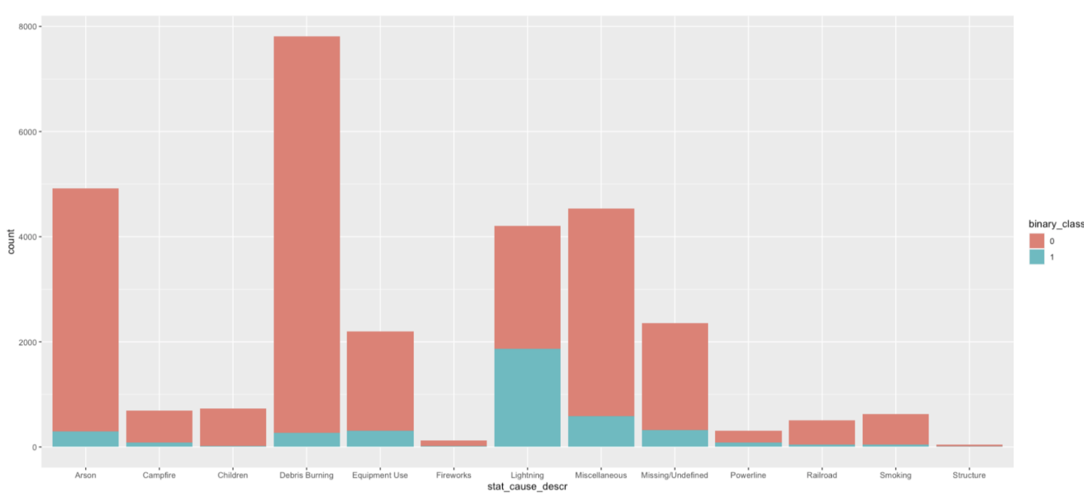

Figure 2: Graph displaying counts of cause of fires, also broken down by the binary response variable (fire size class)

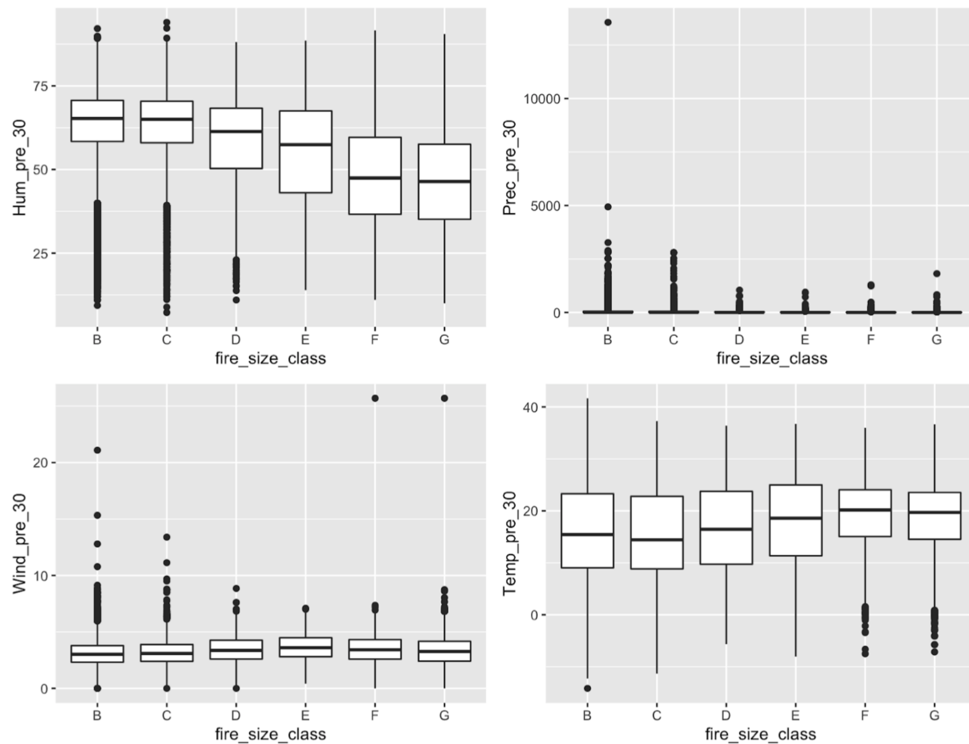

Figure 3: Graphs of meteorological data vs. fire size class

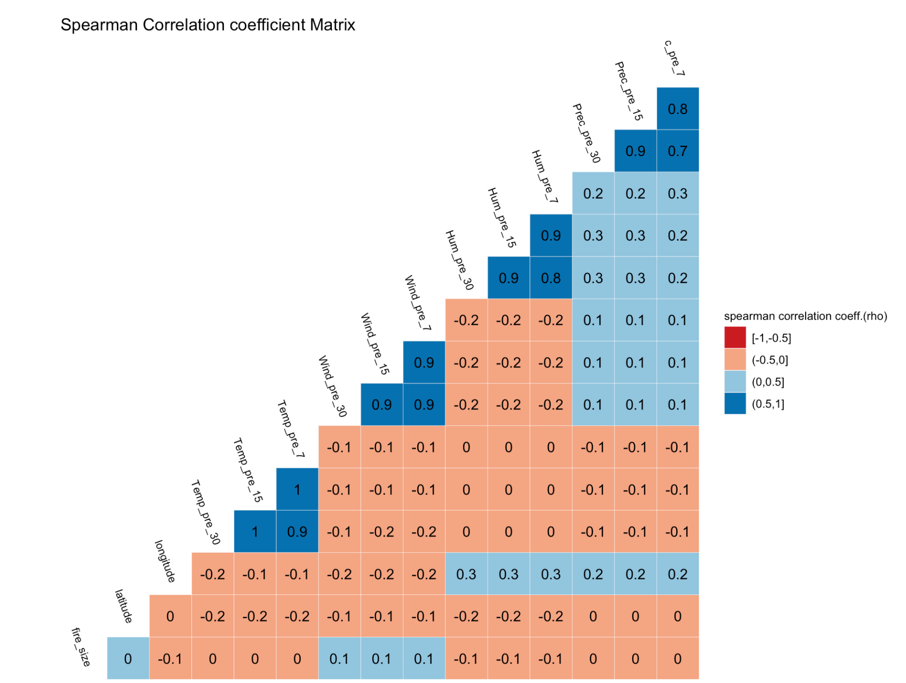

Figure 4: Spearman coefficient correlation matrix

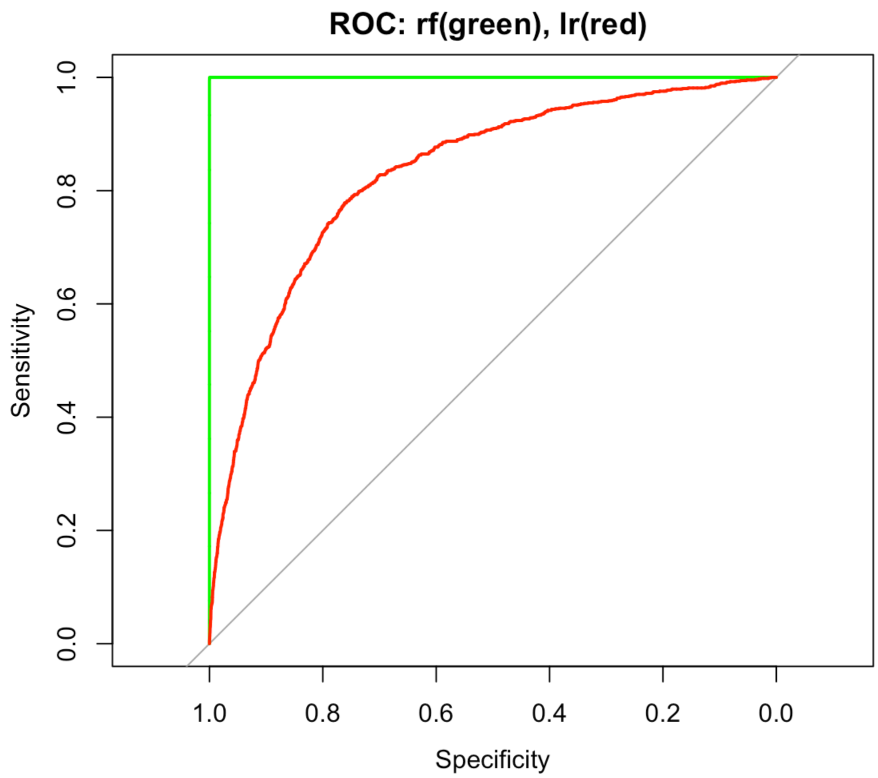

Figure 5: ROC curve

Features

FIRE_SIZE	Estimate of acres within the final perimeter of the fire
FIRE_SIZE_CLASS	Code for fire size based on the number of acres within the final fire perimeter expenditures (A = 0 to 0.25 acres, B = 0.26 to 9.9 acres, C = 10.0 to 99.9 acres, D = 100 to 299 acres, E = 300 to 999 acres, F = 1000 to 4999 acres, and G = 5000+ acres)
STAT_CAUS_DESCR	Cause of fire
LATITUDE	Latitude (NAD83) for point location of the fire (decimal degrees)
LONGITUDE	Longitude (NAD83) for point location of the fire (decimal degrees)
STATE	Two-letter alphabetic code for the state in which the fire burned (or originated), based on the nominal designation in the fire report
DISC_CLEAN_DATE	Date fire was discovered
DISCOVERY_MONTH	Month fire was discovered [rps]
DISC_PRE_YEAR	Calendar year in which the fire was discovered or confirmed to exist
VEGETATION	type of vegetation in the affected area
TEMP/WIND/HUM/PREC_PRE_N where N is 7, 15, 30	Meteorological data N days before the fire was discovered as measured by the closest weather station
REMOTENESS	A measure of how close a fire was from the nearest city (normalized to get a range from 0 to 1)

5 Code

```

# Install packages
install.packages("visdat")
install.packages("corrplot")
install.packages("GGally")
install.packages("gridExtra")
install.packages("naniar")
install.packages("cvTools")
install.packages("glmnet")
install.packages("ISLR")
install.packages("e1071")


# Load packages
library("ggplot2")
library("knitr")
library("visdat")
library("corrplot")
library("GGally")
library("gridExtra")
library("naniar")
library(dplyr)
library("cvTools")
library(tidyverse)
library(caret)
library(randomForest)
library(kernlab)
library(rpart)
library(neuralnet)
library("ISLR")
library("glmnet")
library("e1071")
library(class)

# Loading Data
data <- read.csv("/Users/thibautbadoual/Desktop/MS&E_226_Project/Fire/archive/FW_Veg_Rem_Combined.csv")
# data <- read.csv("~/Docs/Stanford/MSE226/Project/archive/FW_Veg_Rem_Combined.csv")

#----- Data Cleaning ----------------------------------------------------------------------#
# Remove unnecessary columns
data <- subset(data, select = -c(X, Unnamed..0, fire_name, fire_mag,
                                 cont_clean_date, disc_date_final, cont_date_final,
                                 putout_time, disc_date_pre, disc_pre_month,
                                 wstation_usaf, dstation_m, wstation_wban,
                                 wstation_byear, wstation_eyear, weather_file,
                                 Prec_cont, Hum_cont, Wind_cont, Temp_cont,
                                 disc_clean_date, remoteness, disc_pre_year))

# Select only the top ten states
top_ten_states <- group_by(data, state) %>%
  summarise(count = n())
top_ten_states <- top_ten_states[order(-top_ten_states$count),]
top_ten_states <- top_ten_states$state[1:10]
data = filter(data, data$state %in% top_ten_states)

# Change vegetation type
data$Vegetation <- as.character(data$Vegetation)
data$Vegetation[which(data$Vegetation == "4")]  <- "Temperate Evergreen Needleleaf Forest"
data$Vegetation[which(data$Vegetation == "9")]  <- "Grassland/Steppe"
data$Vegetation[which(data$Vegetation == "12")] <- "Open Shrubland"
```

```
data$Vegetation[which(data$Vegetation == "14")] <- "Desert"
data$Vegetation[which(data$Vegetation == "15")] <- "Polar Desert/Rock/Ice"
data$Vegetation[which(data$Vegetation == "16")] <- "Secondary Tropical Evergreen Broadleaf Forest"

# Remove duplicate rows
data <- data[!duplicated(data), ]

# Drop missing values for vegetation data
data = filter(data, data$Vegetation != 0)

# Drop missing values for meteorological data
data = filter(data, data$Prec_pre_30 != -1.00000)
data = filter(data, data$Hum_pre_30 != 0)
data = filter(data, data$Hum_pre_15 != 0)
data = filter(data, data$Hum_pre_7 != 0)

# Add column with binary response variable (fire size class above C = 1; C or below = 0)
data_0 = filter(data, data$fire_size_class %in% c("A", "B", "C"))
data_1 = filter(data, data$fire_size_class %in% c("D", "E", "F", "G"))
data_0$binary_class <- 0
data_1$binary_class <- 1
data <- rbind(data_0, data_1)

#------ Data Overview ------------------------------------------------------------------#
str(data) # Information on columns
summary(data) # Summary of columns
head(data) # Other info on dataset
glimpse(data)
dim(data)

#------ Plots ------------------------------------------------------------------#
# Plots
glimpse(data)

# Number of fires per wildfire class
ggplot(data, aes(fire_size_class)) + geom_histogram(stat = "count") +
  xlab("Wildfire Class (A-G)") + ylab("Number of fire for each class")

# Longitude and Latitude vs fire size class
p1 <- ggplot(data, aes(fire_size_class, longitude)) + geom_boxplot()
p2 <- ggplot(data, aes(fire_size_class, latitude)) + geom_boxplot()
grid.arrange(p1, p2, nrow = 1)

p3 <- ggplot(data, aes(as.factor(discovery_month), Temp_pre_30)) + geom_boxplot() + scale_x_discrete(limits = month.abb)
p4 <- ggplot(data, aes(as.factor(disc_pre_year), Temp_pre_30)) + geom_boxplot()
grid.arrange(p3, p4, nrow = 1)

p5<- ggplot(data, aes(as.factor(discovery_month), Wind_pre_30)) + geom_boxplot() + scale_x_discrete(limits = month.abb)
p6 <- ggplot(data, aes(as.factor(disc_pre_year), Wind_pre_30)) + geom_boxplot()
grid.arrange(p5, p6, nrow = 1)

p7<- ggplot(data, aes(as.factor(discovery_month), Prec_pre_30)) + geom_boxplot() + scale_x_discrete(limits = month.abb)
p8 <- ggplot(data, aes(as.factor(disc_pre_year), Prec_pre_30)) + geom_boxplot()
grid.arrange(p7, p8, nrow = 1)

p9<- ggplot(data, aes(as.factor(discovery_month), Hum_pre_30)) + geom_boxplot() + scale_x_discrete(limits = month.abb)
p10 <- ggplot(data, aes(as.factor(disc_pre_year), Hum_pre_30)) + geom_boxplot()
```

```r
grid.arrange(p9, p10, nrow = 1)

p11 <- ggplot(data, aes(fire_size_class, Hum_pre_30)) + geom_boxplot()
p12 <- ggplot(data, aes(fire_size_class, Prec_pre_30)) + geom_boxplot()
p13 <- ggplot(data, aes(fire_size_class, Wind_pre_30)) + geom_boxplot()
p14 <- ggplot(data, aes(fire_size_class, Temp_pre_30)) + geom_boxplot()
grid.arrange(p11, p12, p13, p14, nrow = 2)

p15 <- ggplot(data, aes(Temp_pre_30)) + geom_histogram(bins = 50)
p16 <- ggplot(data, aes(Prec_pre_30)) + geom_histogram(bins = 50)
p17 <- ggplot(data, aes(Hum_pre_30)) + geom_histogram(bins = 50)
p18 <- ggplot(data, aes(Wind_pre_30)) + geom_histogram(bins = 50)
grid.arrange(p15, p16, p17, p18, nrow = 2)

# Cause of Fire
count_causes_data <- group_by(data, binary_class, stat_cause_descr) %>%
  summarise(count = n())
p19 <- ggplot(count_causes_data, aes(x = stat_cause_descr, y = count)) + geom_bar(position="dodge", stat="identity")
p20 <- ggplot(count_causes_data, aes(x = stat_cause_descr, y = count, fill = binary_class)) + geom_bar(stat="identity")
grid.arrange(p19, nrow = 1)
grid.arrange(p20, nrow = 1)

# State
count_state <- group_by(data, binary_class, state) %>%
  summarise(count = n())
p21 <- ggplot(count_state, aes(x = state, y = count)) + geom_bar(position="dodge", stat="identity")
p22 <- ggplot(count_state, aes(x = state, y = count, fill = binary_class)) + geom_bar(stat="identity")
grid.arrange(p21, nrow = 1)
grid.arrange(p22, nrow = 1)

# Vegetation
count_vegetation <- group_by(data, binary_class, Vegetation) %>%
  summarise(count = n())
p23 <- ggplot(count_vegetation, aes(x = Vegetation, y = count, fill = binary_class)) + geom_bar(position="dodge", stat="identity")
p24 <- ggplot(count_vegetation, aes(x = Vegetation, y = count, fill = binary_class)) + geom_bar(stat="identity")
grid.arrange(p23, nrow = 1)
grid.arrange(p24, nrow = 1)

#######BONUS#######
plot(data$fire_size, data$Hum_pre_30)
plot(data$fire_size, data$Temp_pre_30)
plot(data$fire_size, data$Wind_pre_30)
plot(data$fire_size, data$Prec_pre_30)

plot(data$fire_size, data$longitude)
plot(data$fire_size, data$latitude)

#------ Data Transformations ----------------------------------------------------------------#
# Add column with the seasons instead of the months
data_summer = filter(data, data$discovery_month %in% c("Aug", "Jul", "Sep"))
data_fall = filter(data, data$discovery_month %in% c("Nov", "Oct", "Dec"))
data_winter = filter(data, data$discovery_month %in% c("Feb", "Mar", "Jan"))
data_spring = filter(data, data$discovery_month %in% c("Apr", "May", "Jun"))

data_summer$season <- "summer"
data_fall$season <- "fall"
data_winter$season <- "winter"
data_spring$season <- "spring"
```

```r
data_transformed <- rbind(data_summer, data_fall , data_winter , data_spring)

data_transformed$discovery_month <- NULL
data_transformed$latitude <- NULL

data_transformed$Prec_pre_30 <- log(data_transformed$Prec_pre_30 + 1)
data_transformed$Prec_pre_15 <- log(data_transformed$Prec_pre_15 + 1)
data_transformed$Prec_pre_7 <- log(data_transformed$Prec_pre_7 + 1)
data_transformed$Temp_pre_30 <- log(data_transformed$Temp_pre_30 + 373.15)
data_transformed$Temp_pre_15 <- log(data_transformed$Temp_pre_15 + 373.15)
data_transformed$Temp_pre_7 <- log(data_transformed$Temp_pre_7 + 373.15)

continuous_vars <- c("Prec_pre_30", "Prec_pre_15", "Prec_pre_7",
                     "Temp_pre_30", "Temp_pre_15", "Temp_pre_7",
                     "Wind_pre_30", "Wind_pre_15", "Wind_pre_7",
                     "Hum_pre_30", "Hum_pre_15", "Hum_pre_7")
factor_vars <- c("fire_size", "fire_size_class", "stat_cause_descr",
                 "longitude", "state", "Vegetation", "binary_class", "season")

data_normalized <- cbind(data_transformed[factor_vars], scale(data_transformed[
    continuous_vars]))

#---- Regression Predictions
    ----------------------------------------------------------------------------#

# Set the train et test dataset
set.seed(1)

n = nrow(data);
idx = sample(n, 0.8*n)

data_regression <- data
data_regression_normalized <- data_normalized

data_regression$binary_class <- NULL
data_regression$fire_size_class <- NULL
data_regression_normalized$binary_class <- NULL
data_regression_normalized$fire_size_class <- NULL

train_regression = data_regression[idx ,];
train_regression_normalized = data_regression_normalized[idx ,];
dim(train_regression)

test_regression = data_regression[-idx ,];
test_regression_normalized = data_regression_normalized[-idx ,];
dim(test_regression)

#correlation
ggcorr(data_regression ,
       method = c("all.obs","spearman"),
       nbreaks = 4, palette = 'RdBu', label = TRUE,
       name = "spearman correlation coeff.(rho)",
       hjust = 0.8, angle = -70, size = 3) +
  ggtitle("Spearman Correlation coefficient Matrix")

glimpse(train_regression)
glimpse(test_regression)

#cross validation
ctrl_lm <- trainControl(method = "cv", number = 10)

#Baseline
lmCVFit <- train(fire_size~., data = train_regression, method = "lm",
                 trControl = ctrl_lm, metric = "Rsquared")
print(lmCVFit)
```

```r

#model 1 / all interactions
lmCVFit <- train(fire_size~.+.:., data = train_regression, method = "lm",
                 trControl = ctrl_lm, metric = "Rsquared")
print(lmCVFit)

#model 2 / all transformations
lmCVFit <- train(fire_size~.,
                 data = train_regression_normalized, method = "lm",
                 trControl = ctrl_lm, metric = "Rsquared")
print(lmCVFit)

#model 3 / all relevant interactions
lmCVFit <- train(fire_size~. + Hum_pre_30:Wind_pre_30 + Hum_pre_15:Wind_pre_15 + Hum_pre_7:Wind_pre_7
                 + Hum_pre_30:Temp_pre_30 + Hum_pre_15:Temp_pre_15 + Hum_pre_7:Temp_pre_7
                 + Vegetation:Hum_pre_30 + Vegetation:Wind_pre_30 + Vegetation:Prec_pre_30
                 , data = train_regression, method = "lm",
                 trControl = ctrl_lm, metric = "Rsquared")
print(lmCVFit)

#final model / all combined
lmCVFit <- train(fire_size~. + Hum_pre_30:Wind_pre_30 + Hum_pre_15:Wind_pre_15 + Hum_pre_7:Wind_pre_7
                 + Hum_pre_30:Temp_pre_30 + Hum_pre_15:Temp_pre_15 + Hum_pre_7:Temp_pre_7
                 , data = train_regression_normalized, method = "lm",
                 trControl = ctrl_lm, metric = "Rsquared")
print(lmCVFit)

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

# Compute R^2 from true and predicted values
eval_results <- function(true, predicted, df) {
  SSE <- sum((predicted - true)^2)
  SST <- sum((true - mean(true))^2)
  R_square <- 1 - SSE / SST
  RMSE = sqrt(SSE/nrow(df))


  # Model performance metrics
  data.frame(
    RMSE = RMSE,
    Rsquare = R_square
  )

}

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

#Lasso (alpha = 1) and Ridge (alpha = 0) Regression
X = model.matrix(fire_size ~ 0 + ., train_regression_normalized)
Y = train_regression_normalized$fire_size

train.ind = sample(nrow(X), round(nrow(X)/2))
X.train = X[train.ind,]
X.test = X[-train.ind,]
Y.train = Y[train.ind]
Y.test = Y[-train.ind]

lambdas = 10^seq(-2,3.4,0.1)

# Setting alpha = 0 implements ridge regression
```

```r
fm.ridge <- cv.glmnet(X.train, Y.train, alpha = 0, lambda = lambdas, thresh = 1e-12)
optimal_lambda <- fm.ridge$lambda.min

# Prediction and evaluation on train data
predictions_train <- predict(fm.ridge, s = optimal_lambda, newx = X.train)
eval_results(Y.train, predictions_train, X.train)

# Prediction and evaluation on test data
predictions_test <- predict(fm.ridge, s = optimal_lambda, newx = X.test)
eval_results(Y.test, predictions_test, X.test)


# Setting alpha = 1 implements lasso regression
fm.lasso <- cv.glmnet(X.train, Y.train, alpha = 1, lambda = lambdas, thresh = 1e-12)
optimal_lambda <- fm.lasso$lambda.min

# Prediction and evaluation on train data
predictions_train <- predict(fm.lasso, s = optimal_lambda, newx = X.train)
eval_results(Y.train, predictions_train, X.train)

# Prediction and evaluation on test data
predictions_test <- predict(fm.lasso, s = optimal_lambda, newx = X.test)
eval_results(Y.test, predictions_test, X.test)

#----- Classification Predictions ------------------------------------------------------------------------#

data_predictions2 <- data

data_predictions2$fire_size <- NULL
data_predictions2$fire_size_class <- NULL

train2 = data_predictions2[idx,];
dim(train2)

test2 = data_predictions2[-idx,];
dim(test2)

# build the random forest model and test it
rf_model <- randomForest(binary_class ~., data = train2)
rf_prediction <- predict(rf_model, train2, type = "response")

# build the logistic regression model and test it
lr_model <- glm(binary_class ~., data = train2, family = "binomial")
lr_prediction <- predict(lr_model, train2, type = "response")

# ROC curves
library(pROC)
ROC_rf <- roc(train2$binary_class, rf_prediction)
ROC_lr <- roc(train2$binary_class, lr_prediction)

# Review ROC objects
threshold_rf <- coords(ROC_rf, "best", "threshold")
threshold_lr <- coords(ROC_lr, "best", "threshold")

# Area Under Curve (AUC) for each ROC curve (higher -> better)
ROC_rf_auc <- auc(ROC_rf)
ROC_lr_auc <- auc(ROC_lr)

# plot ROC curves
plot(ROC_rf, col = "green", main = "ROC: rf(green), lr(red) ")

points(ROC_lr, col="red", pch="*")
lines(ROC_lr, col="red")
```

```
# print the performance of each model
paste("Accuracy % of random forest: ", mean(train2$binary_class == round(rf_prediction, digits = 0)))
paste("Accuracy % of logistic regression: ", mean(train2$binary_class == round(lr_prediction, digits = 0)))
paste("Area under curve of random forest: ", ROC_rf_auc)
paste("Area under curve of logistic regression: ", ROC_lr_auc)
paste("Sensitivity of random forest: ", threshold_rf[2])
paste("Sensitivity of logistic regression: ", threshold_lr[2])
```

References

[1] Karen C. Short : Spatial wildfire occurrence data for the united states, 1992-2015 [fpafod20170508]. *4th Edition. Fort Collins, CO: Forest Service Research Data Archive. https://doi.org/10.2737/RDS-2013-0009.4*, 2017.

[2] Noaa national centers for environmental information (2001): Integrated surface hourly [1992-2015]. *ftp://ftp.ncdc.noaa.gov/pub/data/noaa/*.

[3] Prasanth Meiyappan et Atul K. Jain. : "three distinct global estimates of historical land-cover change and land-use conversions for over 200 years.". *Frontiers of Earth Science 6.2: 122-139.*, 2012.

[4] Simplemaps : World cities database. *simplemaps.com/data/world-cities.*