Lab 4

Erica Bishop

2023-02-07

Lab 4: Fire and Tree Mortality

The database we'll be working with today includes 36066 observations of individual trees involved in prescribed fires and wildfires occurring over 35 years, from 1981 to 2016. It is a subset of a larger fire and tree mortality database from the US Forest Service (see data description for the full database here: link). Our goal today is to predict the likelihood of tree mortality after a fire.

Data Exploration

Outcome variable: *yr1status* = tree status (0=alive, 1=dead) assessed one year post-fire.

Predictors: *YrFireName, Species, Genus_species, DBH_cm, CVS_percent, BCHM_m, BTL* (Information on these variables available in the database metadata (link)).

```
trees_dat<- read_csv(file = "https://raw.githubusercontent.com/MaRo406/eds-232-machine-learning/main/da
```

```
## New names:
## Rows: 36066 Columns: 9
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (3): YrFireName, Species, Genus_species dbl (6): ...1, yr1status, DBH_cm,
## CVS_percent, BCHM_m, BTL
## 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.
## * `` -> `...1`
```

Question 1: Recode all the predictors to a zero_based integer form

```
trees_encoded <- recipe(yr1status ~ ., data = trees_dat) |>
  step_integer(all_predictors(), zero_based = TRUE) |>
  prep(trees_dat) |>
  bake(trees_dat) |>
  select(-"...1") #drop index column
```

Data Splitting

Question 2: Create trees_training (70%) and trees_test (30%) splits for the modeling

```
set.seed(123)  # for reproducibility (random sample)
trees_split <- initial_split(trees_encoded, prop = .70)

#create training set
trees_train <- training(trees_split)

#create testing set
trees_test  <- testing(trees_split)
```

Question 3: How many observations are we using for training with this split?

```
print(paste("We are using", nrow(trees_train), "observations for training and ", nrow(trees_test), "for
```

```
## [1] "We are using 25246 observations for training and  10820 for testing."
```

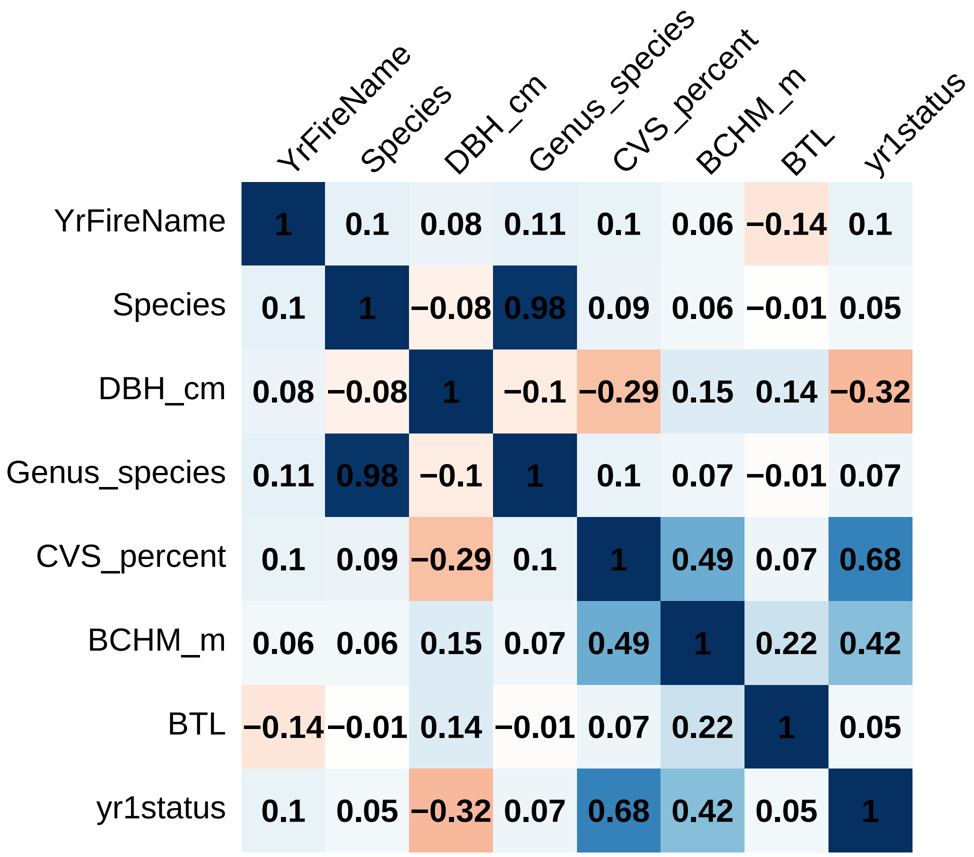
Simple Logistic Regression

Let's start our modeling effort with some simple models: one predictor and one outcome each.

Question 4: Choose the three predictors that most highly correlate with our outcome variable for further investigation.

```
#create a correlation matrix to easily identify variables most highly correlated
cormatrix <- cor(trees_encoded)

# Make a correlation plot between the variables
corrplot(cormatrix, method = "shade", shade.col = NA, tl.col = "black", tl.srt = 45, addCoef.col = "bla
```



	YrFireName	Species	DBH_cm	Genus_species	CVS_percent	BCHM_m	BTL	yr1status
YrFireName	1	0.1	0.08	0.11	0.1	0.06	−0.14	0.1
Species	0.1	1	−0.08	0.98	0.09	0.06	−0.01	0.05
DBH_cm	0.08	−0.08	1	−0.1	−0.29	0.15	0.14	−0.32
Genus_species	0.11	0.98	−0.1	1	0.1	0.07	−0.01	0.07
CVS_percent	0.1	0.09	−0.29	0.1	1	0.49	0.07	0.68
BCHM_m	0.06	0.06	0.15	0.07	0.49	1	0.22	0.42
BTL	−0.14	−0.01	0.14	−0.01	0.07	0.22	1	0.05
yr1status	0.1	0.05	−0.32	0.07	0.68	0.42	0.05	1

The three variables most highly correlated with tree survival are:

- CVS_percent: 0.68
- BCHM_m: 0.42
- DBH_cm: -0.32

CVS_percent is the percent of the pre-fire crown volume that was scorched or consumed by fire (values 0 to 100). If measured, this is the CVS from field measurements. Otherwise it is the calculated CVS from crown length measurement, where CVS=100[(CLS(2CL_pre - CLS))/CL_pre2].

BCHM_m is maximum bark char (also called bole char, bole scorch in other publications) vertical height from ground on a tree bole, rounded to nearest 0.01 m (m=meters).

DBH_cm is the diameter at breast height rounded to nearest 0.1 cm (cm = centimeters).

> Question 5: Use glm() to fit three simple logistic regression models, one for each of the predictors you identified.

```
lm_cvs <- glm(yr1status ~ CVS_percent, family = "binomial", data = trees_train)

lm_bchm <- glm(yr1status ~ BCHM_m, family = "binomial", data = trees_train)

lm_dbh <- glm(yr1status ~ DBH_cm, family = "binomial", data = trees_train)
```

Interpret the Coefficients

We aren't always interested in or able to interpret the model coefficients in a machine learning task. Often predictive accuracy is all we care about.

> Question 6: That said, take a stab at interpreting our model coefficients now.

Interpreting the crown volume model:

```
lm_cvs |>
  broom::tidy()
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic p.value
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)  -6.61     0.111       -59.4       0
## 2 CVS_percent   0.0762   0.00122      62.7       0
```

For the model that regresses tree survival on the percent of tree crown volume burned provides the generalized formula:

$$\text{logit}(\hat{p}) = \log\left(\frac{\hat{p}}{1-\hat{p}}\right) = -6.61 + 0.0762x$$

Therefore, the probabilities can be found with the following formula:

$$\frac{\hat{p}}{1-\hat{p}} = e^{-6.61+0.0762x}$$

To interpret the β_0 coefficient of 0.0762 and the β_1 coefficient of -6.61, we can calculate the probabilities of tree survival with the equation above.

```
exp(coef(lm_cvs)) |>  #exponentiate coeficients for interpretation
  broom::tidy()
```

```
## Warning: 'tidy.numeric' is deprecated.
## See help("Deprecated")
```

```
## # A tibble: 2 x 2
##   names             x
##   <chr>         <dbl>
## 1 (Intercept) 0.00134
## 2 CVS_percent 1.08
```

The intercept (β_0) coefficient tells us that there is a 0.13% chance that a tree will die in one year when 0% of its crown volume is burned.

The CVS_percent coefficient (β_1) tells us that for every 1% increase in crown volume burned, the odds of a tree dying within one year increases multiplicatively by 1.07

Interpreting the bark char height model:

We can follow the same logical steps as outlined above to interpret the model for bark char.

```
lm_bchm |>
  broom::tidy()
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic p.value
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept) -1.98     0.0241       -82.0       0
## 2 BCHM_m       0.00618  0.000104      59.4       0
```

```
exp(coef(lm_bchm)) |>
  broom::tidy()
```

```
## Warning: 'tidy.numeric' is deprecated.
## See help("Deprecated")
```

```
## # A tibble: 2 x 2
##   names           x
##   <chr>       <dbl>
## 1 (Intercept) 0.139
## 2 BCHM_m      1.01
```

The intercept (β_0) coefficient tells us that there is a 13% chance that a tree will die within one year when the maximum height of the bark char is at 0 meters (no bark char).

The BCHM_m coefficient (β_1) tells us that for every 1 meter increase in maximum bar char height, the odds of a tree dying within one year increases multiplicatively by 1.006.

Interpreting the breast-height diameter

```
lm_dbh |>
  broom::tidy()
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)  0.397   0.0282         14.1 6.66e-45
## 2 DBH_cm      -0.00377 0.0000777     -48.4 0
```

```
exp(coef(lm_dbh)) |>
  broom::tidy()
```

```
## Warning: 'tidy.numeric' is deprecated.
## See help("Deprecated")
```

```
## # A tibble: 2 x 2
##   names           x
##   <chr>       <dbl>
## 1 (Intercept) 1.49
## 2 DBH_cm      0.996
```

For the breast-heigth diameter model, the intercept (β_0) coefficient tells us that there is a 148% chance that a tree will die within one year when it's diameter at breast height is 0. If there are trees with a breast-height diameter of 0 (little saplings that are shorter than the measurement) then it makes sense they have a greater likelihood of dying in a fire, but its a less helpful metric to interpret because most trees have some non-zero breast-height diameter.

The BCHM_m coefficient (β_1) tells us that for every 1 centimeter increase in breast-height diameter, the odds of a tree dying within one year decrease multiplicatively by 0.99. Because the coefficient is less than one, it means the odds of survival are increasing.

> Question 7: Now let's visualize the results from these models. Plot the fit to the training data of each model.

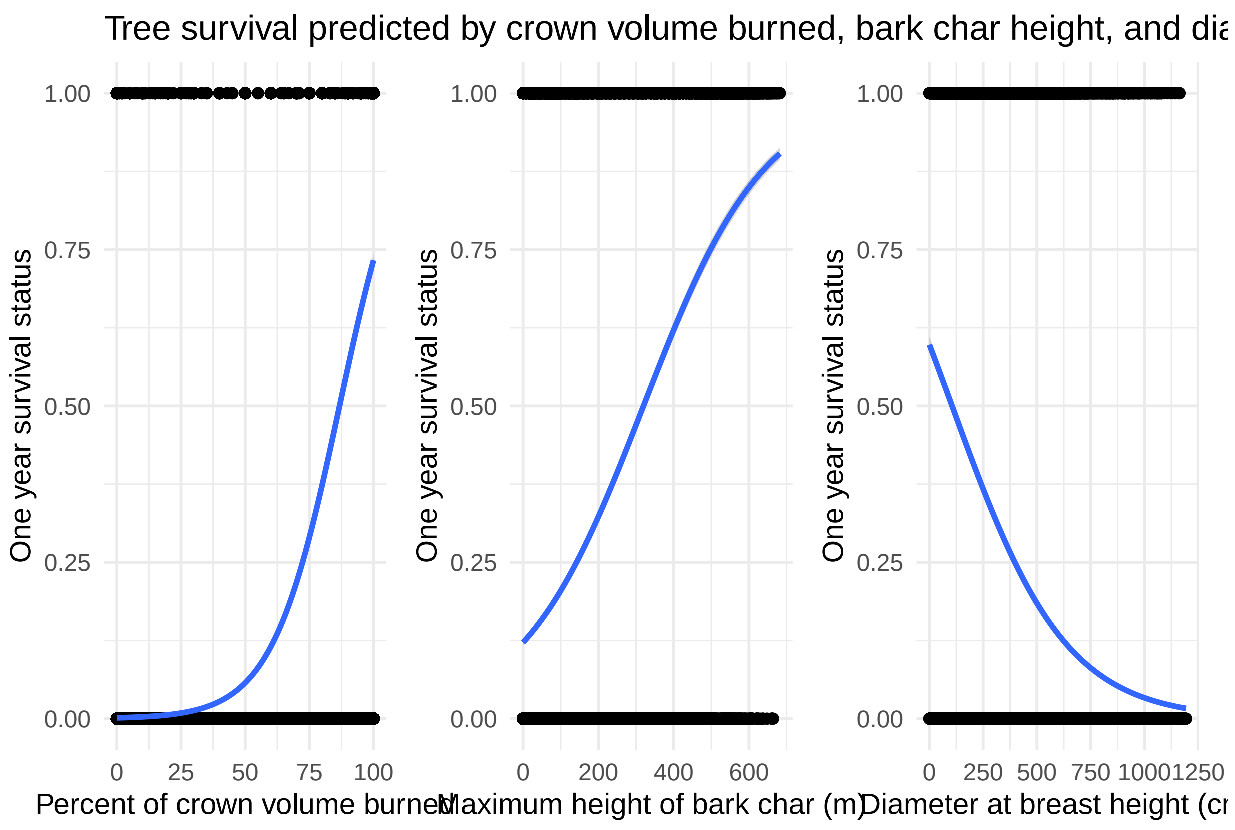
```
cvs_plot <- ggplot(data = trees_train,
                   aes(x = CVS_percent,
                       y = yr1status)) +
  geom_point() +
  stat_smooth(method="glm",  se=TRUE,
              method.args = list(family=binomial)) +
  labs(title = "Tree survival predicted by crown volume burned, bark char height, and diameter",
       x = "Percent of crown volume burned",
       y = "One year survival status") +
  theme_minimal()

bchm_plot <- ggplot(data = trees_train,
                   aes(x = BCHM_m,
                       y = yr1status)) +
  geom_point() +
  stat_smooth(method="glm",  se=TRUE,
              method.args = list(family=binomial)) +
  labs(
       x = "Maximum height of bark char (m)",
       y = "One year survival status") +
  theme_minimal()

dbh_plot <- ggplot(data = trees_train,
                   aes(x = DBH_cm,
                       y = yr1status)) +
  geom_point() +
  stat_smooth(method="glm",  se=TRUE,
              method.args = list(family=binomial)) +
  labs(
       x = "Diameter at breast height (cm)",
       y = "One year survival status") +
  theme_minimal()

cvs_plot + bchm_plot + dbh_plot + patchwork::plot_layout(ncol = 3)
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```



Multiple Logistic Regression

Let's not limit ourselves to a single-predictor model. More predictors might lead to better model performance.

> Question 8: Use glm() to fit a multiple logistic regression called "logistic_full", with all three of the predictors included. Which ofthese are significant in the resulting model?

```
logistic_full <- glm(
  yr1status ~ CVS_percent + BCHM_m + DBH_cm,
  family = "binomial",
  data = trees_train
)

#check out significance

logistic_full |>
  broom::tidy()
```

```
## # A tibble: 4 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept) -5.09     0.114        -44.7 0
## 2 CVS_percent  0.0622   0.00119       52.4 0
## 3 BCHM_m       0.00466  0.000161      28.9 6.05e-184
## 4 DBH_cm      -0.00371  0.000118     -31.4 2.39e-216
```

The p-values for each predictor in this model are near zero, so all are significant.

Estimate Model Accuracy

Now we want to estimate our model's generalizability using resampling.

> Question 9: Use cross validation to assess model accuracy. Use caret::train() to fit four 10-fold cross-validated models (cv_model1, cv_model2, cv_model3, cv_model4) that correspond to each of the four models we've fit so far: three simple logistic regression models corresponding to each of the three key predictors (CVS_percent, DBH_cm, BCHM_m) and a multiple logistic regression model that combines all three predictors.

```r
#make yr1 status a factor
trees_train$yr1status = as.factor(trees_train$yr1status)

set.seed(123)
cv_model1_cvs <- train(
  yr1status ~ CVS_percent,
  data = trees_train,
  method = "glm",
  family = "binomial",
  trControl = trainControl(method = "cv", number = 10)
)

set.seed(123)
cv_model2_bch <- train(
  yr1status ~ BCHM_m,
  data = trees_train,
  method = "glm",
  family = "binomial",
  trControl = trainControl(method = "cv", number = 10)
)

set.seed(123)
cv_model3_dbh <- train(
  yr1status ~ DBH_cm,
  data = trees_train,
  method = "glm",
  family = "binomial",
  trControl = trainControl(method = "cv", number = 10)
)

set.seed(123)
cv_model4_full <- train(
  yr1status ~ CVS_percent + BCHM_m + DBH_cm,
  data = trees_train,
  method = "glm",
  family = "binomial",
  trControl = trainControl(method = "cv", number = 10)
)
```

> Question 10: Use caret::resamples() to extract then compare the classification accuracy for each model. (Hint: resamples() wont give you what you need unless you convert the outcome variable to factor form). Which model has the highest accuracy?

```r
#create a table comparing accuracies of the models
summary(
  resamples(
    list(
      model1 = cv_model1_cvs,
      model2 = cv_model2_bch,
      model3 = cv_model3_dbh,
      model4 = cv_model4_full
    )
  )
)$statistics$Accuracy
```

```
##              Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## model1 0.8899010 0.8923762 0.8962376 0.8975283 0.9006831 0.9080824    0
## model2 0.7588119 0.7658416 0.7717906 0.7714098 0.7758194 0.7837624    0
## model3 0.7437624 0.7475003 0.7534165 0.7522385 0.7555446 0.7603960    0
## model4 0.8902101 0.8969307 0.9037624 0.9031131 0.9093792 0.9144216    0
```

Model 4, the multiple logistic regression model, has the highest accuracy across all quantiles.

Let's move forward with this single most accurate model.

> Question 11: Compute the confusion matrix and overall fraction of correct predictions by the model.

```
#predict classes with the model
pred_death <- predict(cv_model4_full, trees_train)

#show confusion matrix and accuracy
confusionMatrix(data = relevel(pred_death, ref = 1),
                reference = relevel(trees_train$yr1status, ref =  1))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 16504   847
##          1  1595  6300
##
##                Accuracy : 0.9033
##                  95% CI : (0.8996, 0.9069)
##     No Information Rate : 0.7169
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.769
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9119
##             Specificity : 0.8815
##          Pos Pred Value : 0.9512
##          Neg Pred Value : 0.7980
##              Prevalence : 0.7169
##          Detection Rate : 0.6537
##    Detection Prevalence : 0.6873
##       Balanced Accuracy : 0.8967
##
##        'Positive' Class : 0
##
```

Question 12: Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.

0 represents a tree that has survived and 1 represents a tree that has died. So the confusion matrix shows predicted values in the columns, column 1 predicting tree survival and column 2 predicting tree death. The rows show the actual outcomes, row 1 is tree survival and row 2 is tree death.

Therefore, the model accurately predicted tree survival for 16504 of the observations across resamples, and accurately predicted tree death for 6300 of the observations across resamples. The model falsely predicted death for 847 observations (false positive) and falsely predicted survival (false negative) for 1595 observations.

Question 13: What is the overall accuracy of the model? How is this calculated?

The overall accuracy of the model is 90.33%. This is calculated based on how much better the model is than using the majority class classifier (i.e. predicingting tree survival for all observatons). According to the documentation for `caret::confusionMatrix`, a 95% confidence interval and a one-sided test statistic is used to compute accuracy.

Test Final Model

Alright, now we'll take our most accurate model and make predictions on some unseen data (the test data).

Question 14: Now that we have identified our best model, evaluate it by running a prediction on the test data, trees_test.

```r
#make test data yr1status a factor
trees_test$yr1status = as.factor(trees_test$yr1status)

#predict classes with the test data
pred_test <- predict(cv_model4_full, trees_test)

#assess the accuracy with the test predictions
#show confusion matrix and accuracy
confusionMatrix(data = relevel(pred_test, ref = 1),
                reference = relevel(trees_test$yr1status, ref =  1))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 7013  362
##          1  721 2724
##
##                Accuracy : 0.8999
##                  95% CI : (0.8941, 0.9055)
##     No Information Rate : 0.7148
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7628
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9068
##             Specificity : 0.8827
##          Pos Pred Value : 0.9509
##          Neg Pred Value : 0.7907
##              Prevalence : 0.7148
##          Detection Rate : 0.6482
##    Detection Prevalence : 0.6816
##       Balanced Accuracy : 0.8947
##
##        'Positive' Class : 0
##
```

Question 15: How does the accuracy of this final model on the test data compare to its cross validation accuracy? Do you find this to be surprising? Why or why not?

The model preformed similarly on the test data as it did to the cross-validated training data, 89.99% overall accuracy compaared to 90.33% overall accuracy. This makes sense because our training data set was more extensive and therefore gave the model a good representation of observations. Both models are statistically significant, and fall within the same 95% confidence interval.