Homework 6

Jules Merigot (8488256)

November 21, 2022

PSTAT 131/231 Statistical Machine Learning - Fall 2022

Tree-Based Models

Exercise 1

Before we get started, let's load the Pokemon data in into our workspace.

```
Pokemon_data <- read.csv(file = "C:/Users/jules/OneDrive/Desktop/homework-5/data/Pokemon.csv")
```

Let's load the janitor package, and use its `clean_names()` function on the Pokémon data. We'll save the results to work with for the rest of the assignment.

```
library(janitor)

Pokemon_data <- Pokemon_data %>%
  clean_names()
```

For this assignment, we'll handle the rarer classes by simply filtering them out. Let's filter the entire data set to contain only Pokemon whose `type_1` is Bug, Fire, Grass, Normal, Water, or Psychic.

```
Pokemon_data <- Pokemon_data %>%
  filter(grepl("Bug|Fire|Grass|Normal|Water|Psychic", type_1))
```

Now that we're done filtering, let's convert `type_1`, `legendary`, and `generation` to factors.

```
Pokemon_data$type_1 <- factor(Pokemon_data$type_1)
Pokemon_data$legendary <- factor(Pokemon_data$legendary)
Pokemon_data$generation <- factor(Pokemon_data$generation)
```

Let's perform an initial split of the data, and stratify by the outcome variable.

```
set.seed(8488)

Pokemon_split <- initial_split(Pokemon_data, prop=0.70, strata=type_1)

Pokemon_train <- training(Pokemon_split)
Pokemon_test <- testing(Pokemon_split)
```

For splitting the data, I chose a proportion of 0.70 because it allows for more training data, while retaining enough data to be tested since there is a limited amount of observations. The training data has 559 observations while the testing data has 241 observations.

Next, let's use v-fold cross-validation on the training set, using 5 folds. We'll stratify the folds by `type_1` as well.

```
Pokemon_folds <- vfold_cv(Pokemon_train, v = 5, strata=type_1)
```

In this case, stratifying the folds is useful to ensure that each fold is representative of all strata of the data.

Let's set up a recipe to predict `type_1` with `legendary`, `generation`, `sp_atk`, `attack`, `speed`, `defense`, `hp`, and `sp_def`. We'll also dummy-code `legendary` and `generation`, as well as center and scale all predictors.

```
Pokemon_recipe <- recipe(type_1 ~ legendary + generation + sp_atk + attack +
                           speed + defense + hp + sp_def, data=Pokemon_train) %>%
  step_dummy(c(legendary, generation)) %>%
  step_normalize(all_predictors())

Pokemon_recipe %>% prep() %>% juice()
```

```
## # A tibble: 318 x 13
##    sp_atk attack   speed defense      hp   sp_def type_1 legen~1 gener~2 gener~3
##     <dbl>  <dbl>   <dbl>   <dbl>   <dbl>    <dbl> <fct>    <dbl>   <dbl>   <dbl>
##  1 -1.62  -1.36  -0.820   -1.18  -0.850  -1.82    Bug     -0.245  -0.416  -0.486
##  2  0.522 -0.886  0.0241  -0.649 -0.310   0.365   Bug     -0.245  -0.416  -0.486
##  3 -1.62  -1.20  -0.652   -1.35  -1.03   -1.82    Bug     -0.245  -0.416  -0.486
##  4 -1.47  -1.51  -1.16    -0.649 -0.850  -1.63    Bug     -0.245  -0.416  -0.486
##  5 -0.857  0.525  0.193   -1.00  -0.130   0.365   Bug     -0.245  -0.416  -0.486
##  6 -1.78   2.40   2.56    -1.00  -0.130   0.365   Bug     -0.245  -0.416  -0.486
##  7 -0.857 -0.102 -1.50    -0.472 -1.21   -0.544   Bug     -0.245  -0.416  -0.486
##  8  0.522 -0.259  0.700   -0.296  0.0505  0.183   Bug     -0.245  -0.416  -0.486
##  9 -0.550  1.15   1.21     0.410  0.0505  0.365   Bug     -0.245  -0.416  -0.486
## 10 -0.550  1.62   0.531    1.12  -0.130   0.00126 Bug     -0.245  -0.416  -0.486
## # ... with 308 more rows, 3 more variables: generation_X4 <dbl>,
## #   generation_X5 <dbl>, generation_X6 <dbl>, and abbreviated variable names
## #   1: legendary_True, 2: generation_X2, 3: generation_X3
```