Code

Homework 5

PSTAT 131/231

Homework 5

For this assignment, we will be working with the file `"pokemon.csv"`, found in `/data`. The file is from Kaggle: https://www.kaggle.com/abcsds/pokemon (https://www.kaggle.com/abcsds/pokemon).

The Pokémon (https://www.pokemon.com/us/) franchise encompasses video games, TV shows, movies, books, and a card game. This data set was drawn from the video game series and contains statistics about 721 Pokémon, or "pocket monsters." In Pokémon games, the user plays as a trainer who collects, trades, and battles Pokémon to (a) collect all the Pokémon and (b) become the champion Pokémon trainer.

Each Pokémon has a primary type (https://bulbapedia.bulbagarden.net/wiki/Type) (some even have secondary types). Based on their type, a Pokémon is strong against some types, and vulnerable to others. (Think rock, paper, scissors.) A Fire-type Pokémon, for example, is vulnerable to Water-type Pokémon, but strong against Grass-type.

Fig 1. Vulpix, a Fire-type fox Pokémon from Generation 1 (also my favorite Pokémon!)

The goal of this assignment is to build a statistical learning model that can predict the **primary type** of a Pokémon based on its generation, legendary status, and six battle statistics. *This is an example of a* ***classification problem****, but these models can also be used for* ***regression problems***.

Read in the file and familiarize yourself with the variables using `pokemon_codebook.txt`.

Exercise 1

Install and load the `janitor` package. Use its `clean_names()` function on the Pokémon data, and save the results to work with for the rest of the assignment. What happened to the data? Why do you think `clean_names()` is useful?

Homework 5

For 231 Students

Exercise 2

Using the entire data set, create a bar chart of the outcome variable, `type_1`.

How many classes of the outcome are there? Are there any Pokémon types with very few Pokémon? If so, which ones?

For this assignment, we'll handle the rarer classes by grouping them, or "lumping them," together into an 'other' category. Using the `forcats` package (https://forcats.tidyverse.org/), determine how to do this, and **lump all the other levels together except for the top 6 most frequent** (which are Bug, Fire, Grass, Normal, Water, and Psychic).

Convert `type_1` and `legendary` to factors.

Exercise 3

Perform an initial split of the data. Stratify by the outcome variable. You can choose a proportion to use. Verify that your training and test sets have the desired number of observations.

Next, use *v*-fold cross-validation on the training set. Use 5 folds. Stratify the folds by `type_1` as well. *Hint: Look for a `strata` argument.*

Why do you think doing stratified sampling for cross-validation is useful?

Exercise 4

Create a correlation matrix of the training set, using the `corrplot` package. *Note: You can choose how to handle the categorical variables for this plot; justify your decision(s).*

What relationships, if any, do you notice?

Exercise 5

Set up a recipe to predict `type_1` with `legendary`, `generation`, `sp_atk`, `attack`, `speed`, `defense`, `hp`, and `sp_def`.

- Dummy-code `legendary` and `generation`;
- Center and scale all predictors.

Exercise 6

We'll be fitting and tuning an elastic net, tuning `penalty` and `mixture` (use `multinom_reg()` with the `glmnet` engine).

Set up this model and workflow. Create a regular grid for `penalty` and `mixture` with 10 levels each; `mixture` should range from 0 to 1. For this assignment, let `penalty` range from 0.01 to 3 (this is on the `identity_trans()` scale; note that you'll need to specify these values in base 10 otherwise).

Homework 5

For 231 Students

Exercise 7

Now set up a random forest model and workflow. Use the `ranger` engine and set `importance = "impurity"`; we'll be tuning `mtry`, `trees`, and `min_n`. Using the documentation for `rand_forest()`, explain in your own words what each of these hyperparameters represent.

Create a regular grid with 8 levels each. You can choose plausible ranges for each hyperparameter. Note that `mtry` should not be smaller than 1 or larger than 8. **Explain why neither of those values would make sense.**

What type of model does `mtry = 8` represent?

Exercise 8

Fit all models to your folded data using `tune_grid()`.

Note: Tuning your random forest model will take a few minutes to run, anywhere from 5 minutes to 15 minutes and up. Consider running your models outside of the .Rmd, storing the results, and loading them in your .Rmd to minimize time to knit. We'll go over how to do this in lecture.

Use `autoplot()` on the results. What do you notice? Do larger or smaller values of `penalty` and `mixture` produce better ROC AUC? What about values of `min_n`, `trees`, and `mtry`?

What elastic net model and what random forest model perform the best on your folded data? (What specific values of the hyperparameters resulted in the optimal ROC AUC?)

Exercise 9

Select your optimal **random forest model** in terms of `roc_auc`. Then fit that model to your training set and evaluate its performance on the testing set.

Using the **training** set:

- Create a variable importance plot, using `vip()`. *Note that you'll still need to have set `importance = "impurity"` when fitting the model to your entire training set in order for this to work.*
    - What variables were most useful? Which were least useful? Are these results what you expected, or not?

Using the testing set:

- Create plots of the different ROC curves, one per level of the outcome variable;
- Make a heat map of the confusion matrix.

Exercise 10

How did your best random forest model do on the testing set?

Which Pokemon types is the model best at predicting, and which is it worst at? (Do you have any ideas why this might be?)

For 231 Students

Exercise 11

In the 2020-2021 season, Stephen Curry, an NBA basketball player, made 337 out of 801 three point shot attempts (42.1%). Use bootstrap resampling on a sequence of 337 1's (makes) and 464 0's (misses). For each bootstrap sample, compute and save the sample mean (e.g. bootstrap FG% for the player). Use 1000 bootstrap samples to plot a histogram of those values. Compute the 99% bootstrap confidence interval for Stephen Curry's "true" end-of-season FG% using the quantile function in R. Print the endpoints of this interval.

Exercise 12

Using the `abalone.txt` data from previous assignments, fit and tune a **random forest** model to predict `age`. Use stratified cross-validation and select ranges for `mtry`, `min_n`, and `trees`. Present your results. What was your final chosen model's **RMSE** on your testing set?