Data Science for Public Policy

Aaron R. Williams - Georgetown University

PPOL 670 | Assignment 06
Supervised Machine Learning

Due Date: Sunday, November 7th at 11:59 PM.

Deliverable:

1. An .Rmd file with your R code
2. The resulting project .html file
3. The URL of a private Git repository.

Grading Rubric

Please show your work! It is easier to give partial credit when a computational mistake is made if formulas are fully specified and substitutions are correctly made.

- [1 point] Create a private, well-managed GitHub repository, including an appropriate `.gitignore`, and an informative `README.md` file. You should add at least one commit for each question. Points will be reduced for infrequent commits or unclear commit messages.
- [1 point] Write a clean and well-composed `.Rmd file`, including separate named code chunks for each task required below. The resulting `.html` file should show code and results, but hide unnecessary warnings and messages.
- [1 point] Exercise 01
- [1 point] Exercise 02
- [0.5 point] Exercise 03
- [1 point] Exercise 04
- [1.5 points] Exercise 05
- [2 points] Exercise 06

Points: 9 points

Learning and data science are both collaborative practices. We encourage you to discuss class topics and homework topics with each other. However, the work you submit must be your own. A student should never see another student's code or receive explicit coding instructions for a homework problem. Please attend office hours or contact one of the instructors if you need help or clarification.

Plagiarism on homework or projects will be dealt with to the full extent allowed by Georgetown policy (see http://honorcouncil.georgetown.edu).

Setup

Create a new folder with a new R project (`.Rproj`) and R Markdown file (`.Rmd`). Then create a new **private GitHub repository**. Add awunderground and ncstabile17 to the private GitHub repository.

Exercise 01 (1 point)

Calculate the mean square error (MSE), root mean square error (RMSE), and mean absolute error (MAE) for the following data "by hand". You can add scanned answers with `knitr::include_graphics()` or you can use inline LaTeX equations with R Markdown. How do RMSE and MAE handle outlier predictions differently?

true_value	predicted_value
1	2
2	2
3	1
4	8
5	4

Exercise 02 (1 point)

The following data come from a binary classification problem.

true_value	predicted_value
0	0
0	0
0	1
0	0
0	0
1	1
1	0
1	0
1	1
1	1

Using the above data, calculate the following "by hand" and show your work:

1. A confusion matrix
2. Accuracy
3. Precision
4. Recall/Sensitivity

You can scan your paper answer and add it with `knitr::include_graphics()` or you can use Markdown tables. Do not use `yardstick::conf_mat()` or `caret::confusionMatrix()`.

Exercise 03 (0.5 point)

The following data come from a multiclass classification problem.

true_value	predicted_value
compliance	compliance
compliance	compliance
compliance	compliance
compliance	risk of noncompliance
compliance	compliance
compliance	noncompliance
compliance	compliance
compliance	compliance
compliance	compliance
compliance	risk of noncompliance
risk of noncompliance	risk of noncompliance
risk of noncompliance	noncompliance
noncompliance	noncompliance
noncompliance	compliance
noncompliance	noncompliance

Using the above data, calculate the following "by hand" and show your work:

1. A confusion matrix
2. Accuracy
3. Misclassification rate

You can scan your paper answer and add it with `knitr::include_graphics()` or you can use Markdown tables. Do not use `yardstick::conf_mat()` or `caret::confusionMatrix()`.

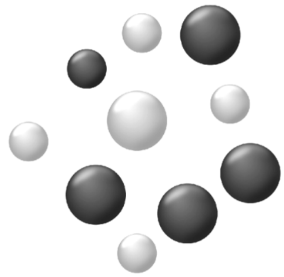
Exercise 04 (1 point)

Consider a population where it is known that 0.49 of observations have a value of `0` and 0.51 of observations have a value of `1`. Approximately what accuracy can be achieved by simply guessing the same value for all observations? What number should you predict?

Consider a population where it is known that 0.99 of observations have a value of `0` and 0.01 of observations have a value of `1`. Approximately what accuracy can be achieved by simply guessing the same value for all observations? What number should you predict?

Explain why it is important to consider context when comparing calculated accuracy in different supervised machine learning tasks?

Exercise 05 (1.5 points)



`marbles.csv` contains a new simple random sample from the population of marbles that generated the first machine learning example in class #7.

1. Divide the marbles data set into a training set with 80% of observations and a testing set 20% of observations. Set the seed to `20200229` before sampling.
2. Use `count()` and `library(ggplot2)` to develop and justify a intuitive/mental model for predicting black marbles.
3. Construct a custom function that takes a vector of sizes and returns a vector of predicted colors. Apply it to the testing data. The R4DS chapter on functions is helpful.
4. Construct a custom function that takes `y` and `y_hat` that returns calculated accuracy and a confusion matrix. Until now, we have only returned one object from a custom function. Use `list()` inside of `return()` to return more than one object. Apply it to the data from part 3. Do not use `yardstick::conf_mat()` or `caret::confusionMatrix()`.
5. Using the same testing and training data, estimate a decision tree/CART model with functions from `library(parnsip)`. Use the "rpart" engine.
6. Does the decision tree/CART model generate the same predictions on the testing data as the model from part 2? Why or why not?

Exercise 06 (2 points)

The following example includes a simulated data set about the presence of rat burrows in alleys and proximity to the nearest jumbo slice pizza restaurant. (rats.R is on Canvas)

- `rat_burrow` 1 if burrow present, 0 if no burrow present
- `pizza_distance` Distance in miles from the alley to the nearest jumbo slice pizza restaurant

The goal is to estimate a K-Nearest Neighbors model "by hand" with three different Ks. Run the following code chunk to create three resamples of the data.

Note: *running the code out-of-order will change the observations included in reach resample. Run the entire code chunk for consistent results.*

```
set.seed(20200302)

# input the data
rats <- tribble(
  ~rat_burrow, ~pizza_distance,
  1, 0.01,
  1, 0.05,
  1, 0.08,
  0, 0.1,
```

```
  0, 0.12,
  1, 0.2,
  1, 0.3,
  1, 0.5,
  1, 0.75,
  0, 0.9,
  1, 1,
  0, 1.2,
  0, 2.2,
  0, 2.3,
  0, 2.5,
  1, 3,
  0, 3.5,
  0, 4,
  0, 5,
  0, 7
) %>%
  mutate(rat_burrow = factor(rat_burrow))

# split into training and testing data
split <- initial_split(rats, prop = 0.75)
rats_training <- training(split)
rats_testing <- testing(split)

rats_k1 <- vfold_cv(data = rats_training,
                    v = 3)

rats_k3 <- vfold_cv(data = rats_training,
                    v = 3)

rats_kn <- vfold_cv(data = rats_training,
                    v = 3)
```

Extract the analysis data and assessment data from the first resample in `rats_k1`, `rats_k3`, and `rats_kn` with `analysis()` and `assessment()`. **Hint:** You can access the first resample for the first problem with `rats_k1$splits[[1]]`. The observations should slightly differ in each resample.

- Calculate `y_hat` for the assessment data "by hand" in the first resample of `rats_k1` with KNN and $k = 1$.
- Calculate `y_hat` for the assessment data "by hand" in the first resample of `rats_k3` with KNN and $k = 3$.
- Calculate `y_hat` for the assessment data "by hand" in the first resample of `rats_kn` with KNN and $k = n$.

Note: Only make the calculations for the first resample. This is to save time!

You can write non-`library(tidymodels)` code or arithmetic to come up with $\hat{y}$. In each case, add `y_hat` to the assessment data using `bind_cols()`. Include the data frame in your R Markdown document using `knitr::kable()`. Calculate accuracy and a confusion matrix using your function from the marbles exercise.

Which model was easiest to estimate computationally and why? Which model was toughest to estimate computationally and why?