Data Science for Public Policy

Aaron R. Williams - Georgetown University

PPOL 670 | Assignment 07
Machine Learning

Due Date: Sunday, November 21st at 11:59 PM.

Deliverable:

1. A .Rmd file with your R code.
2. The resulting project .html file.
3. The URL to your private GitHub repository from assignment 06.

Grading Rubric

- [3 points] Exercise 01
- [3 points] Exercise 02
- [6 points] Exercise 03

Points: 12 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

Add a second `.Rmd` to the GitHub repository from assignment06.Rmd.

Exercise 01 (3 points)

This exercise will use restaurant inspections data from the NYC OpenData Portal by way of tidytuesday. The grades can be A, B, or C but this exercise will work with "A" or "Not A" so that this is a binary classification problem. Use the following code to create the data set for this application. (also included as a .R script)

```
library(tidyverse)
library(lubridate)

# use this url to download the data directly into R
df <- read_csv("https://data.cityofnewyork.us/api/views/43nn-pn8j/rows.csv")

# clean names with janitor
sampled_df <- df %>%
  janitor::clean_names()

# create an inspection year variable
sampled_df <- sampled_df %>%
  mutate(inspection_date = mdy(inspection_date)) %>%
  mutate(inspection_year = year(inspection_date))

# get most-recent inspection
sampled_df <- sampled_df %>%
  group_by(camis) %>%
  filter(inspection_date == max(inspection_date)) %>%
  ungroup()

# subset the data
sampled_df <- sampled_df %>%
  select(camis, boro, zipcode, cuisine_description, inspection_date,
         action, violation_code, violation_description, grade,
         inspection_type, latitude, longitude, council_district,
         census_tract, inspection_year, critical_flag) %>%
  filter(complete.cases(.)) %>%
  filter(inspection_year >= 2017) %>%
  filter(grade %in% c("A", "B", "C"))

# create the binary target variable
sampled_df <- sampled_df %>%
  mutate(grade = if_else(grade == "A", "A", "Not A")) %>%
  mutate(grade = as.factor(grade))

# create extra predictors
sampled_df <- sampled_df %>%
  group_by(boro, zipcode, cuisine_description, inspection_date,
         action, violation_code, violation_description, grade,
         inspection_type, latitude, longitude, council_district,
         census_tract, inspection_year)  %>%
  mutate(vermin = str_detect(violation_description, pattern = "mice|rats|vermin|roaches")) %>%
  summarize(violations = n(),
            vermin_types = sum(vermin),
            critical_flags = sum(critical_flag == "Y")) %>%
  ungroup()

# write the data
write_csv(sampled_df, "restaurant_grades.csv")
```

1. Estimate a Model

a. Split the data into training and testing sets with the seed `20201020`.
b. Create a recipe with (at least) `themis::step_downsample(grade)`. Downsampling is one technique to deal with class imbalances.
c. Estimate a decision tree for classification with the "rpart" engine. There is no need to use resampling methods like v-fold cross-validation because you are only estimating one model.

2. Evaluate the Model

a. Create a confusion matrix using your model estimated on the training data and the testing data.
b. Calculate the precision and recall/sensitivity "by hand". "By hand" can use R code, please just show your work and don't use the tidymodels functions.
c. Calculate the precision and recall/sensitivity with functions from `library(tidymodels)`. Compare your answers with b.
d. Describe the quality of the model.

3. Improvement

a. Describe ideas for improving the model with 4 or 5 sentences. Improvements can include different feature engineering, different algorithms, different hyperparameters, and adding auxiliary information.

4. Variable Importance

Measures of variable importance can be calculated with most types of predictive models. Max Kuhn offers a brief outline here. The authors of `library(rpart)` briefly explain the method for the rpart model on page 11 here

`library(vip)` info here and `library(tidymodels)` can quickly create a plot of the variable importance. The following code plots the ten most important predictions from a fit workflow.

```
library(vip)

rpart_fit %>%
  pull_workflow_fit() %>%
  vip(num_features = 10)
```

a. Briefly explain how this measure is calculated and interpret the results.

5. Application

a. How might this model be used by an ombudsman working for the NYC health department?

Exercise 02 (3 points)

The objective of this exercise is to predict Chicago "L" train ridership at the Clarke/Lake station. The data come from Feature and Target Engineering by Max Kuhn and Kjell Johnson. I recommend reading the description. After loading `library(tidymodels)`, run the following to set up the data.

```
Chicago_estimation <- Chicago %>%
  slice(1:5678)

Chicago_implementation <- Chicago %>%
  slice(5679:5698) %>%
  select(-ridership)
```

1. Convert date into a useable variable

In this application, unaltered date won't be useful for prediction because all new observations will have unseen dates by definition. Use `library(lubridate)` to generate the following before splitting your data:

- `weekday` is the day of the week. Use `label = TRUE` so the value is a factor
- `month` is the month of the year. Use `label = TRUE` so the value is a factor.
- `yearday` is the numeric day number of the year.

2. Set up a testing environment

a. Split `Chicago_estimation` into a training set and a testing set with seed `20211101`. Use the default `prop`.
b. Demonstrate some exploratory data analysis with the training data.
c. Set up v-fold cross-validation with 10 folds.

3. Test different approaches

Use cross-validation to compare at least three different models that meet the following conditions:

1. Estimate one parametric regression model (linear regression, LASSO, etc.), a random forest model, and a model of your choice (e.g. KNN, decision/regression tree, MARS).
2. Use hyperparameter tuning for at least one of your models.
3. Use at least three `step_*()` functions from `library(recipes)`. You must use `step_holiday()`.
4. Calculate the MAE and RMSE for each resample. Plot the RMSE across the 10 resamples for the three (or more) different models. Find the average RMSE of each model across the 10 resamples (this should result in three (or more) numbers).

4. Estimate the out-of-sample error rate

a. Train the best model on the full training data (without resampling).
b. Make predictions with the testing data and calculate the RMSE. This is your out-of-sample error rate estimate. You only want to touch the testing data once!

5. Implement the final model

Using your "best" model from the cross validation, estimate the ridership for the 20 observations in `Chicago_implementation`. Be sure to print your 20 predictions so they are visible in your submission. You only want to touch the implementation data once!

6. Briefly describe your final model

Write a few sentences describing your final model. Is it a good model? Is it globally interpretable? Is it locally interpretable? What are the most important predictors?

Note: There are two options for exercise 03!

This exercise is meant to be involved and it should be an opportunity to start the final project. Please reach out early and often. Please reach out if you have an alternative idea.

Exercise 03 Supervised ML Option (6 points)

It is acceptable to end up with a model that is not useful! Your grade depends on your evaluation of the usefulness–not the creation of a perfect model.

1. Set up

a. Find a government/policy-relevant data set with a practical regression or classification application. You can use the data you have already found/used for your project of interest. Briefly describe the data set. Clearly state the prediction application and describe the target variable.
b. Using data visualization and other skills from this course, perform an exploratory data analysis (on the training data!). During this process, perform necessary data cleaning (please reach out if the data cleaning is significant work but you wish to use this data for your project).
c. Explicitly pick an error metric. Explicitly state the costs of an error. How much error is too much if you use a regression model? Is a false positive or a false negative more costly if you use a classification problem?

2. Come up with Models

a. Describe three model specifications. In the three specifications:
   - State the predictor variables included in the model and necessary preprocessing for each variable
   - Use at least least two different types of preprocessing
   - Use at least two different algorithms (i.e. linear regression, KNN, CART, random forest).
   - Use at least one algorithm with hyperparameters. Your first specification should be an algorithm with a hyperparameter(s).

3. Estimation

a. Implement the first candidate model on the training data with resampling methods and hyperparameter tuning using `library(tidymodels)`.

4. Interpretation

a. Interpret the results in the context of the application.
b. Write at least three sentences explaining why you think your second specification will be better or worse than your first specification.
c. Write at least three sentences explaining why you think your third specification will be better or worse than your first specification.
d. Based on the previous two bullets, suggest a specification that you think will outperform your 2nd and 3rd specifications. Your answer can include ways that you would change the modeling process, preprocess the data, and/or add auxiliary information from other data sets.
e. Thinking back to the results from your first specification, explain why your model is useful or not considering implementation, the error rate, and context.

Exercise 03 Unsupervised ML Option (6 points)

It is acceptable to end up with a model that is not useful! Your grade depends on your evaluation of the usefulness–not the creation of a perfect model.

1. Set up

a. Find a government/policy-relevant data set with a practical clustering application. You can use the data you have already found/used for your project of interest. Briefly describe the data set. Clearly state the clustering application, state the number of clusters you think will be useful/practical, and describe the predictors of interest.
b. Using data visualization and other skills from this course, perform an exploratory data analysis. During this process, perform necessary data cleaning (please reach out if the data cleaning is significant work but you wish to use this data for your project).

2. Come up with Models

a. The scale and correlation of predictors is very important when clustering. Appropriately preprocess your predictors.
b. Briefly outline what your predictors are measuring and why they are appropriate for this analysis. Avoid redundant predictors.
c. Describe three clustering model specifications. In the three specifications:
    - State the predictor variables included in the mode
    - Use at least two different algorithms (e.g. K-Means, DBCSAN).

3. Estimation

a. Implement the first clustering model in R.
b. Pick the optimal number of clusters for the data set using subject matter expertise, WSS, average silhouette width, and the gap statistic. Justify your choice.

4. Interpretation

a. Interpret the results in the context of the application.
b. Write at least three sentences explaining why you think your second specification will be better or worse than your first specification. This will require an understanding of the different clustering algorithms.
c. Write at least three sentences explaining why you think your third specification will be better or worse than your first specification.
d. Based on the previous two bullets, suggest a specification that you think will outperform your 2nd and 3rd specifications. Your answer can include ways that you would change the modeling process, preprocess the data, and/or add auxiliary information from other data sets.
e. Thinking back to the results from your first specification, explain why your model is useful or not considering implementation and context.