MATH 154 - HW7 - Using Recipes

Ra-Zakee Muhammad

due: Thursday, Oct 28, 2021

summary

The assignment will give practice for both understanding what a recipe is and for implementing recipes with the idea of model building.

requisites

Read chapter 8 of *Tidy Modeling with R*, https://www.tmwr.org/recipes.html.

assignment

1. **Pod Q** Describe one thing you learned from someone in your pod this week (it could be: content, logistical help, background material, R information, something fun, etc.) 1-3 sentences.
2. Multiple Choice questions.

(a) Consider the `Credit` dataset in the **ISLR** package. Which of the following are true statements?

```r
library(ISLR)
recipe(Balance ~ Income + Rating + Cards, data = Credit)
```

i. `Balance` is the only predictor
ii. All variables of the data frame are used as predictors, except for `Balance`.
iii. `Balance` is the outcome (or response) variable. (True)
iv. There are three predictors. (True)

(b) What is true about a `recipe()`?

i. A recipe on its own is a collection of steps, it does not change the dataset. (TRUE)
ii. A formula defines the dataset to be used.
iii. The `.` sign (e.g., `Balance ~ .`) in a recipe formula represents *all* of the variables in the dataset.

(c) Which of the following statements about the preprocessing step are correct?

i. The required preprocessing steps vary depending on the use case. (True)
ii. Preprocessing transforms the variables into an appropriate format for use in later steps of the workflow. (TRUE)
iii. Preprocessing steps might be adjusted several times during the machine learning workflow.
iv. Preprocessing means to build a baseline model, based on which we can adjust settings.

3. Look up the following `step_` functions. What do they do?

(a) `step_other`

groups all unique catagorical observations whose counts make up less than a thresholds percentage of the total observations into a single category called other.

(b) `step_dummy`

changes a n long column of categorical data into (m) n long columns where m is the number of unique catagorical observations in the original n long column and the entries in these columns are either 0 or 1 where we have a 1 in a column and row if that row in the original column is the name of the column in the new column.

(c) `step_normalize`

noormalize numeric data to have a sd of 1 and mean of 0.

4. In R, type `ls(pattern = '^all_', env = as.environment('package:tidymodels'))`. (The output should be six selectors).

Describe the six selectors.

```r
ls(pattern = '^all_', env = as.environment('package:recipes'))
```

```
## [1] "all_nominal"            "all_nominal_predictors" "all_numeric"
## [4] "all_numeric_predictors" "all_outcomes"           "all_predictors"
```

1) all catagorical and text variables
2) goes through predictor variables and chooses all catagorical and text variables ones
3) all numeric variables
4) goes through predictor variables and chooses all all_numeric variables
5) Goes through outcome variables in formula and selects them
6) Goes through predictor variables in formula and selects them

5. The `recipe()` has two main parts: a formula and a dataset.

(a) Which part (the formula or the dataset) distinguishes between predictor (explanatory) and outcome (response) variables?

the formula

(b) Which part (the formula or the dataset) distinguishes between numeric (quantitative) and nominal (categorical, factor, string) variables?

the dataset

6. In this problem, we will build a linear model to predict the average sale price of a home in Ames, IA. The dataset (called `ames`) lives in the **tidymodels** package. The outcome variable will be the log10 transformed `Sale_Price`.

```
library(tidymodels)
data(ames)
ames <- ames %>% mutate(Sale_Price = log10(Sale_Price))
```

(a) Partition the data into two groups, a training set and a test set. The training data should include 80% of the observations.

```
set.seed(4747)
ames_split <- initial_split(ames, prop = .5)
ames_train <- training(ames_split)
ames_test  <-  testing(ames_split)
```

(b) Build a recipe using the following predictor variables:

- the neighborhood
- above ground living area
- year built
- building type
- latitude
- longitude

In your recipe perform the following preprocessing `step_*`s:

- transform the living area using log base 10
- keep only the neighborhoods which represent at least 1% of the observations
- create dummy variables from all the nominal predictors

```
ames_rec <-
  recipe(Sale_Price ~ Neighborhood + Gr_Liv_Area + Year_Built + Bldg_Type +
           Latitude + Longitude, data = ames_train) %>%
  step_log(Gr_Liv_Area, base =10) %>%
  step_other(Neighborhood, threshold = .01) %>%
  step_dummy(all_nominal_predictors())
```

(c) Create a model object to run a linear regression, set the engine to "lm".

```
lm_model <-linear_reg() %>%
 set_engine("lm")
```

(d) Create a workflow object. Start with workflow, then add the model and add the recipe.

```
lm_wflow <-
  workflow() %>%
  add_model(lm_model) %>%
  add_recipe(ames_rec)
```

(e) Fit the training data to the linear model that was created by the workflow.

```
lm_fit <- lm_wflow %>%
  fit(data = ames_train)
```

(f) Using the **broom** package, `tidy()` the linear model fit to see the coefficients.

```
library(broom)
lm_fit %>% tidy()
```

```
## # A tibble: 30 x 5
##    term                                estimate   std.error statistic       p.value
##    <chr>                                  <dbl>       <dbl>     <dbl>         <dbl>
##  1 (Intercept)                     -9.702128e+1 3.708716e+1 -2.616034 8.989108e-  3
##  2 Gr_Liv_Area                      6.348201e-1 1.773162e-2 35.80159  3.948362e-201
##  3 Year_Built                       2.022338e-3 1.414176e-4 14.30047  1.813408e- 43
##  4 Latitude                         4.873985e-1 4.562845e-1  1.068190 2.856144e-  1
##  5 Longitude                       -8.093037e-1 3.581175e-1 -2.259883 2.397791e-  2
##  6 Neighborhood_College_Creek      -3.404087e-2 2.920419e-2 -1.165616 2.439634e-  1
##  7 Neighborhood_Old_Town           -3.317986e-2 1.125181e-2 -2.948846 3.241251e-  3
##  8 Neighborhood_Edwards            -7.414085e-2 2.284042e-2 -3.246037 1.197236e-  3
##  9 Neighborhood_Somerset            2.445756e-2 1.476427e-2  1.656537 9.783178e-  2
## 10 Neighborhood_Northridge_Heights  9.773915e-2 1.878302e-2  5.203590 2.238090e-  7
## # ... with 20 more rows
```