Tidymodels Workflow Example

Will Doyle

10/28/2021

This guide will provide the quickest "through line" when using workflows.

```
library(tidyverse)
```

```
## Warning: replacing previous import 'lifecycle::last_warnings' by
## 'rlang::last_warnings' when loading 'pillar'

## Warning: replacing previous import 'lifecycle::last_warnings' by
## 'rlang::last_warnings' when loading 'hms'

## Warning: package 'tibble' was built under R version 4.1.2
```

```
library(tidymodels)
```

```
## Warning: package 'recipes' was built under R version 4.1.2
```

```
mv_df<-read_rds("mv.Rds")%>%
  filter(!is.na(budget))%>%
  mutate(log_gross=log(gross))
```

Example 1: Simple linear regression

First, we'll split the data into training and testing sets:

```
split_data<-initial_split(mv_df)

mv_train<-training(split_data)

mv_test<-testing(split_data)
```

Define Model

Next, let's define the model we want to use: OLS regression

```
lm_fit <-
  linear_reg() %>%
  set_engine("lm")%>%
  set_mode("regression")
```

Define Recipe

A recipe is a set of steps that gets the data ready for analysis. A recipe requires a formula that defines the outcomes and the predictors, and a dataset. We'll do two: take the log of budget, and create a series of dummy variables for rating.

```
mv_formula<-as.formula("log_gross~budget+rating")

mv_recipe<-recipe(mv_formula,mv_train)%>%
  step_log(budget)%>%
  step_dummy(rating)
```

Create workflow

A workflow contains all of these steps, allowing us to put them together in a unified framework. Our workflow will start with the model and the recipe.

```
movie_wf<-workflow()%>%
  add_model(lm_fit)%>%
  add_recipe(mv_recipe)
```

Fit workflow to training data

Now, we can run this model on our training data, establishing the relationship between our predictors (budget and rating) and the outcome (log of gross).

```
movie_wf<-movie_wf%>%
  fit(mv_train)
```

```
## Warning: There are new levels in a factor: NA
```

Check model fit in testing data

We want to check our predictions against the testing dataset. Since we have the full testing dataset in hand, we can do it using last_fit.

```
movie_lf<-last_fit(movie_wf,split_data)
```

```
## Warning: package 'rlang' was built under R version 4.1.2

## Warning: package 'vctrs' was built under R version 4.1.2

## ! train/test split: preprocessor 1/1: There are new levels in a factor: NA

## ! train/test split: preprocessor 1/1, model 1/1 (predictions): There are new levels in a fac...
```

```
movie_lf$.metrics
```

```
## [[1]]
## # A tibble: 2 x 4
##   .metric .estimator .estimate .config
##   <chr>   <chr>          <dbl> <chr>
## 1 rmse    standard       1.28  Preprocessor1_Model1
## 2 rsq     standard       0.540 Preprocessor1_Model1
```

For your exercise, you'll need to use the testing dataset we gave you, which will work like this:

```
mv_test<-
  movie_wf%>%
  predict(new_data=mv_test)%>%
  bind_cols(mv_test)
```

```
## Warning: There are new levels in a factor: NA
```

```
mv_test
```

```
## # A tibble: 798 x 22
##    .pred title    rating genre  year released score  votes director writer star
##    <dbl> <chr>    <chr>  <chr> <dbl> <chr>    <dbl>  <dbl> <chr>    <chr>  <chr>
##  1  16.8 America~ R      Come~  2000 April 1~   7.6 5.14e5 Mary Ha~ Bret ~ Chri~
##  2  17.0 Memento  R      Myst~  2000 May 25,~   8.4 1.2 e6 Christo~ Chris~ Guy ~
##  3  19.7 The Per~ PG-13  Acti~  2000 June 30~   6.4 1.6 e5 Wolfgan~ Sebas~ Geor~
##  4  19.1 The Pat~ R      Acti~  2000 June 28~   7.2 2.6 e5 Roland ~ Rober~ Mel ~
##  5  18.4 Erin Br~ R      Biog~  2000 March 1~   7.4 1.82e5 Steven ~ Susan~ Juli~
##  6  19.2 Unbreak~ PG-13  Drama  2000 Novembe~   7.3 3.96e5 M. Nigh~ M. Ni~ Bruc~
##  7  17.5 Road Tr~ R      Come~  2000 May 19,~   6.4 1.61e5 Todd Ph~ Todd ~ Brec~
##  8  18.4 Traffic  R      Crime  2000 January~   7.6 1.99e5 Steven ~ Simon~ Mich~
##  9  19.4 Gone in~ PG-13  Acti~  2000 June 9,~   6.5 2.65e5 Dominic~ H.B. ~ Nico~
## 10  17.8 Pitch B~ R      Acti~  2000 Februar~   7.1 2.29e5 David T~ Jim W~ Radh~
## # ... with 788 more rows, and 11 more variables: country <chr>, budget <dbl>,
## #   gross <dbl>, company <chr>, runtime <dbl>, id <dbl>, imdb_id <chr>,
## #   bechdel_score <dbl>, boxoffice_a <dbl>, language <chr>, log_gross <dbl>
```

The `mv_test` file now contains a new prediction for ever case in the testing dataset.