3.modelling (exercises)

Joao Lopes

2024-10-07

Contents

1. BUILD A MODEL

1.1. LOOK AT DATA

[no exercises]

1.2. FIT A MODEL

a) Evaluate the following model fitted to the diamonds datasets using native R function summary(). Evaluate it also using function augment() and rsq() from tidymodels. What are the possible problems of using the same data to fit and to evaluate the model? Think about when you have completely new data that you want to make predictions of.

```
set.seed(123)
diamonds_data <- diamonds |>
  mutate(
    log_price = log(price),
    log_carat = log(carat),
    fct_cut = factor(cut, ordered = FALSE),
    .keep = "used"
  ) |>
  slice_sample(n = 1000)
lm_fit <- linear_reg() |>
  fit(log_price ~ log_carat + fct_cut, data = diamonds_data)
```

b) How can you avoid over estimating the evaluation of the model? Consider taking a new sample of the data.

c) Using the same fitted model, build by hand a plot showing how the residuals vary a long the values of the predictors.

1.3. USE MODEL TO PREDICT

a) Using the model from the previous section, predict the price of diamonds with 1, 2 and 3 carat for cuts "Fair", "Very Good" and "Ideal". Build a plot of the predictions using a confidence interval of 90%.

---

2. BUILD A MODEL USING WORKFLOW

2.1. LOOK AT DATA

[no exercises]

2.2. SPLIT DATA

[no exercises]

2.3. CREATE WORKFLOW

a) Considering the following workflow, build a new workflow without the steps of the recipe. Look at the function add_formula().

```
flights_rec <-
  recipe(arr_delay ~ ., data = train_data) |>
  step_date(date, features = c("dow", "month")) |>
  step_holiday(date, holidays = timeDate::listHolidays("US")) |>
  step_rm(date) |>
  step_dummy(all_nominal_predictors()) |>
  step_zv(all_predictors()) |>
  step_normalize(all_numeric_predictors())
flights_mod <- logistic_reg(engine = "glm")
flights_wflow <-
  workflow() |>
  add_model(flights_mod) |>
  add_recipe(flights_rec)
```

2.4. FIT A MODEL

a) Using the given workflow and the newly created one, fit the two models to the following data set. Don't forget to split the data in train and test sets.

```
set.seed(123)
flight_data <- flights |>
  slice_sample(n = 10000) |>
  mutate(
    arr_delay = factor(ifelse(arr_delay >= 30, "late", "on_time")),
    date = lubridate::as_date(time_hour)
  ) |>
  select(dep_time, origin, dest, distance, carrier, date, arr_delay) |>
  na.omit() |>
  mutate(across(where(is.character), as.factor))
set.seed(456)
data_split <- initial_split(flight_data, prop = 3/4)
train_data <- training(data_split)
test_data  <- testing(data_split)
```

2.5. EVALUATE MODEL

a) How did the previously fitted models perform? Use the appropriate statistic.

b) Calculate a confusion matrix by hand to the model that considers the steps of the recipe in the workflow. Which probability threshold was used for predicting the classes?

c) Use probability threshold for predicting the classes of 0.40 and 0.60. How do the threshold impact on the confusion matrix?

d) According to the confusion matrix the fitted model as a sensitivity of 0.120 and a specificity of 0.990. Using ggplot, place these values in a ROC curve. Use geom_path(), geom_abline(), coord_equal() and theme_bw().

e) Find which terms are more important to the fitted model and plot them. Use the statistics of each term as a proxy for their importance. Try also using the exp-transformed estimates of the coefficients. Are the results similar? Note that term "(Intercept)" is meaningless and should be discarded.

---

3. BUILD VARIOUS MODELS USING WORKFLOW

3.1. LOOK AT DATA

[no exercises]

3.2. SPLIT DATA

[no exercises]

3.3. CREATE WORKFLOW AND FIT MODEL 1

a) Considering the following split data set and tidymodel recipe, create the respective workflow and fit a simple glm model. Use a ROC curve and the AUC to evaluate your model.

```
hotels <-
  read_csv("https://tidymodels.org/start/case-study/hotels.csv") |>
  mutate(across(where(is.character), as.factor))
set.seed(123)
splits      <- initial_split(hotels, strata = children, prop = 0.75)
hotel_other <- training(splits)
hotel_test  <- testing(splits)
set.seed(234)
val_set <- validation_split(hotel_other, strata = children, prop = 0.80)
holidays <- c("AllSouls", "AshWednesday", "ChristmasEve", "Easter",
              "ChristmasDay", "GoodFriday", "NewYearsDay", "PalmSunday")
lr_recipe <-
  recipe(children ~ ., data = hotel_other) |>
  step_date(arrival_date) |>
  step_holiday(arrival_date, holidays = holidays) |>
  step_rm(arrival_date) |>
  step_dummy(all_nominal_predictors()) |>
  step_zv(all_predictors()) |>
  step_normalize(all_predictors())
```

3.4. CREATE WORKFLOW AND FIT MODEL 2

a) Replace the glm model used above with a neural network model. Consider function mlp() with engine "nnet" and mode "classification". For the hyper parameters, set hidden_units = 1 and penalty = tune(). Use the same recipe as before. To tune the model set a grid with 10 points and use the following data split. Choose the best model and evaluated with a ROC curve and the AUC. How confident are you about the tuning?.

```
set.seed(234)
val_set <- validation_split(hotel_other, strata = children, prop = 0.80)
```

3.5. EVALUATE MODEL

a) Choose the best neural network model and evaluated it using the test data created before. For the evaluation, consider the ROC curve and the AUC. Finish the analysis by plotting the most important features (i.e. predictors) of the final model. Set the importance parameter of the engine to "impurity".