Class Activity 23

Your name here

February 25 2024

Group Activity 1

Load the `mlbench` package to get `PimaIndiansDiabetes2` dataset.

```
# Load the data - diabetes
data(PimaIndiansDiabetes2)
db <- PimaIndiansDiabetes2
db <- db %>% drop_na()
db_raw <- db %>% select(glucose, insulin, diabetes)

db_split <- initial_split(db_raw, prop = 0.80)
# Create training data
db_train <- db_split %>% training()
# Create testing data
db_test <- db_split %>% testing()
```

a. *Creating the Recipe:* Construct a recipe for the model by normalizing `glucose` and `insulin` predictors to predict `diabetes` status on the training set, ensuring data scales are comparable.

```
db_recipe <- recipe(_____________, data = ________) %>%
  step_scale(all_predictors()) %>%
  step_center(all_predictors()) %>%
  prep()
Error: <text>:1:22: unexpected input
1: db_recipe <- recipe(__
                        ^
```

b. *Model Specification:* Define the KNN model using a flexible `tune()` placeholder for the number of neighbors, specifying a `classification` task.

```
knn_spec <- nearest_neighbor(weight_func = "rectangular",
                             engine = "kknn",
                             mode = "classification",
                             neighbors = _______)
Error: <text>:4:43: unexpected input
3:                              mode = "classification",
4:                              neighbors = __
                                            ^
```

c. ***Creating Folds:*** **Divide the training data into 10 stratified folds based on the diabetes outcome to prepare for cross-validation, ensuring representation.**

```
db_vfold <- vfold_cv(_______, v = _________, strata = ________, repe)
Error: <text>:1:23: unexpected input
1: db_vfold <- vfold_cv(__
                        ^
```

d. ***Cross-Validation Grid:*** **Generate a sequence of K values to test with 10-fold cross-validation, evaluating model performance across a range of neighbors.**

```
k_vals <- tibble(neighbors = _____________________)
Error: <text>:1:31: unexpected input
1: k_vals <- tibble(neighbors = __
                                ^
```

```
knn_fit <- workflow() %>%
  add_recipe(______) %>%
  add_model(_______) %>%
  tune_grid(
    resamples = ______,
    grid = _____,
    metrics = metric_set(yardstick::ppv, yardstick::accuracy, sens, spec),
    control = control_resamples(save_pred = TRUE))
Error: <text>:2:15: unexpected input
1: knn_fit <- workflow() %>%
2:   add_recipe(__
                 ^
```

```
cv_metrics <- collect_metrics(_____)
Error: <text>:1:32: unexpected input
1: cv_metrics <- collect_metrics(__
                                 ^
```

e. ***Visualization:*** **Plot the cross-validation results to determine the optimal K value, comparing different performance metrics visually.**

```
final.results <- cv_metrics %>%  mutate(.metric = as.factor(.metric)) %>%
  select(neighbors, .metric, mean)
Error in eval(expr, envir, enclos): object 'cv_metrics' not found

final.results %>%
  ggplot(aes(x = neighbors, y = mean, color = forcats::fct_reorder2(.metric, neighbors, mean))) +
  geom_line(size = 1) +
  geom_point(size = 2) +
  theme_minimal() +
  scale_color_wsj() +
  scale_x_continuous(breaks = k_vals[[1]]) +
  theme(panel.grid.minor.x = element_blank())+
  labs(color='Metric', y = "Estimate", x = "K")
Error in eval(expr, envir, enclos): object 'final.results' not found
```

Group Activity 2

a. Data Preparation and Train-Test Split

Load the `mlbench` package and `tidymodels` framework, select relevant features for predicting `glucose`, and split the data into training and test sets. For this activity, use `mass` and `insulin` as your features.

```
library(mlbench)
library(tidymodels)
library(dplyr)

data(PimaIndiansDiabetes2)
db <- PimaIndiansDiabetes2 %>%
  drop_na() %>%
  select()

# Splitting the data
set.seed(2056)
db_split <- initial_split(db, prop = 0.75)
db_train <- training(db_split)
db_test <- testing(db_split)
```

b. Model Specification

Define a linear regression model for predicting `glucose` as a function of `mass` and `insulin`.

```
lm_spec <-

lm_spec
Error in eval(expr, envir, enclos): object 'lm_spec' not found
```

c. Fit the Model

Fit the linear model to the training data, predicting `glucose` based on `mass` and `insulin`.

```
lm_mod <-

Error: <text>:3:0: unexpected end of input
1: lm_mod <-
2:
  ^
```

d. Predict on Test Data and Evaluate the Model

Use the fitted model to predict `glucose` levels on the test set and evaluate the model's accuracy with RMSE and R-squared metrics.

```
# Predicting glucose levels
results <- db_test %>%
  bind_cols(predictions = predict(lm_mod, new_data = , type = )) %>%
  select( )
Error: object 'lm_mod' not found

# Displaying first 6 predictions
results %>%
  slice_head(n = 6) %>%
  knitr::kable()
```

```
Error in eval(expr, envir, enclos): object 'results' not found
```

```
# Evaluating the model
eval_metrics <- metric_set(rmse, rsq)

eval_metrics(data = ,
             truth = ,
             estimate = ) %>%
  select(-2) %>%
  knitr::kable()
Error in `metric_set()`:
! Failed to compute `rmse()`.
Caused by error:
! argument "data" is missing, with no default
```

(Bonus) Create a scatter plot to visualize the actual vs. predicted glucose levels, including a regression line for reference.

```
results %>%
  ggplot(aes(x = , y = )) +
  geom_point(color = "blue", alpha = 0.6) +
  geom_smooth(method = "lm", color = "red", linetype = "dashed") +
  labs(title = "Predicted vs Actual Glucose Levels",
       x = "Actual Glucose",
       y = "Predicted Glucose") +
  theme_minimal()
Error in eval(expr, envir, enclos): object 'results' not found
```