BST 219
Core Principles of Data Science

Lecture 25: Machine Learning continuing
December 3, 2024

Recipe of the Day!

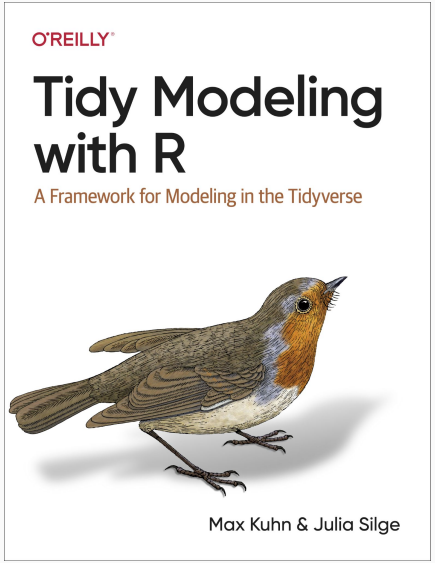
[Simply the Best Blueberry Pie]
(Credit to Kenny)

Happy December!

Agenda

- Announcements
  - Lab this week!
  - Homework #4 due December 16th by 11:59pm
  - Final projects have been assigned (see Canvas announcement for repo access) and TFs have been assigned
- Continue Machine Learning module

Tidy Modeling with R



Melanoma example

```r
library(tidyverse)
library(caret)
library(dslabs)
library(boot)

data(melanoma, package ="boot") # Melanoma example

melanoma = melanoma %>% subset(status %in% c(1, 2)) %>% # Change 1 and 2 values to "Dead" and "Alive"
  mutate(status = as.factor(ifelse(status == 1, "Dead", "Alive")))

y <- melanoma$status # Save outcome as y
x <- melanoma$thickness # Save predictor as x

set.seed(10) # For reproducibility
train_index <- createDataPartition(y, times = 1, p = 0.5, list = FALSE) # Partition data

train_set <- melanoma[train_index, ] # Create training set
test_set <- melanoma[-train_index, ] # Create test set
```

- Load data and create train and test sets
- Use the **same** train and test sets for **all models**

ML Method: Logistic Regression

```r
# Fit logistic regression model
glm_fit <- train_set %>%
  mutate(y = as.numeric(status == "Dead")) %>%
  glm(y ~ thickness, data = ., family = "binomial")

# Calculate predicted probabilities
p_hat_logit <- predict(glm_fit, newdata = test_set, type="response")

# Create predictions of "Dead" or "Alive" using 0.5 probability cutoff
# (if predicted probability is > 0.5, predict "Dead")
y_hat_logit <- ifelse(p_hat_logit > 0.5, "Dead", "Alive")

# Print coonfusion matrix to view evaluation metrics
confusionMatrix(data = as.factor(y_hat_logit),
                reference = test_set$status, positive = "Dead")
```

ML Method: Naive Bayes

```r
library(e1071)

# Fit Naive Bayes model
nb_fit  <- naiveBayes(status ~ thickness, data = train_set)

# Calculate predicted probabilities
p_hat_nb <- predict(nb_fit, test_set, type = "raw")[,2]

# Create predictions of "Dead" or "Alive" using 0.5 probability cutoff (default)
# (if predicted probability is > 0.5, predict "Dead")
y_hat_nb <- predict(nb_fit, test_set)

# Print coonfusion matrix to view evaluation metrics
confusionMatrix(data = as.factor(y_hat_nb),
                reference = test_set$status, positive = "Dead")
```

ML Method: K Nearest Neighbors (kNN)

```
# Fit a knn model (default k is k = 5)
knn_fit <- knn3(status ~ thickness, data = train_set)

# Calculate predicted probabilities
p_hat_knn <- predict(knn_fit, newdata = test_set)[,2]

# Create predictions of "Dead" or "Alive" using 0.5 probability cutoff (default)
# (if predicted probability is > 0.5, predict "Dead")
y_hat_knn <- ifelse(p_hat_knn > 0.5, "Dead", "Alive")

# Print coonfusion matrix to view evaluation metrics
confusionMatrix(data = as.factor(y_hat_knn),
                reference = test_set$status, positive = "Dead")
```

ML Algorithms Summary Table

Machine Learning Algorithms Summary Table

Algorithm	Binary Classification	Multiclass Classification	Regression	Advantages	Disadvantages
Logistic regression `glm(family = "binomial")`	✓			Very common /understood Can calculate feature importance (which predictors are the most predictive)	Parameter estimates can be unstable when a lot of separation between classes Lower performance compared to other models when predictors are normally distributed in each of the classes Limited to binary classification (can use multinomial logistic regression for multiclass classification)
Naive Bayes `naiveBayes()`	✓	✓		Not as strong of assumptions as QDA and LDA Reduction in variance	Can be biased
kNN `knn3()`	✓	✓	✓	Nonparametric (can lead to better performance)	The sample size needs to be much larger than the number of predictors Does not indicate which predictors are important