DATA 622 Assignment 2

CUNY: Spring 2021

Philip Tanofsky

09 March 2021

Introduction

```
# Import required R libraries
library(palmerpenguins)
library(tidyverse)
library(caret)
library(MASS)
library(ggplot2)
library(mvtnorm)
theme_set(theme_classic())
```

```
ds <- penguins

head(ds)
```

```
## # A tibble: 6 x 8
##   species island bill_length_mm bill_depth_mm flipper_length_~ body_mass_g sex
##   <fct>   <fct>           <dbl>         <dbl>            <int>       <int> <fct>
## 1 Adelie  Torge~           39.1          18.7              181        3750 male
## 2 Adelie  Torge~           39.5          17.4              186        3800 fema~
## 3 Adelie  Torge~           40.3          18                195        3250 fema~
## 4 Adelie  Torge~           NA            NA                 NA          NA <NA>
## 5 Adelie  Torge~           36.7          19.3              193        3450 fema~
## 6 Adelie  Torge~           39.3          20.6              190        3650 male
## # ... with 1 more variable: year <int>
```

```
summary(ds)
```

```
##       species          island    bill_length_mm  bill_depth_mm
##  Adelie   :152   Biscoe   :168   Min.   :32.10   Min.   :13.10
##  Chinstrap: 68   Dream    :124   1st Qu.:39.23   1st Qu.:15.60
##  Gentoo   :124   Torgersen: 52   Median :44.45   Median :17.30
##                                  Mean   :43.92   Mean   :17.15
##                                  3rd Qu.:48.50   3rd Qu.:18.70
##                                  Max.   :59.60   Max.   :21.50
##                                  NA's   :2       NA's   :2
##  flipper_length_mm  body_mass_g       sex           year
```

```
##  Min.   :172.0     Min.   :2700   female:165   Min.   :2007
##  1st Qu.:190.0     1st Qu.:3550   male  :168   1st Qu.:2007
##  Median :197.0     Median :4050   NA's  : 11   Median :2008
##  Mean   :200.9     Mean   :4202                Mean   :2008
##  3rd Qu.:213.0     3rd Qu.:4750                3rd Qu.:2009
##  Max.   :231.0     Max.   :6300                Max.   :2009
##  NA's   :2         NA's   :2
```

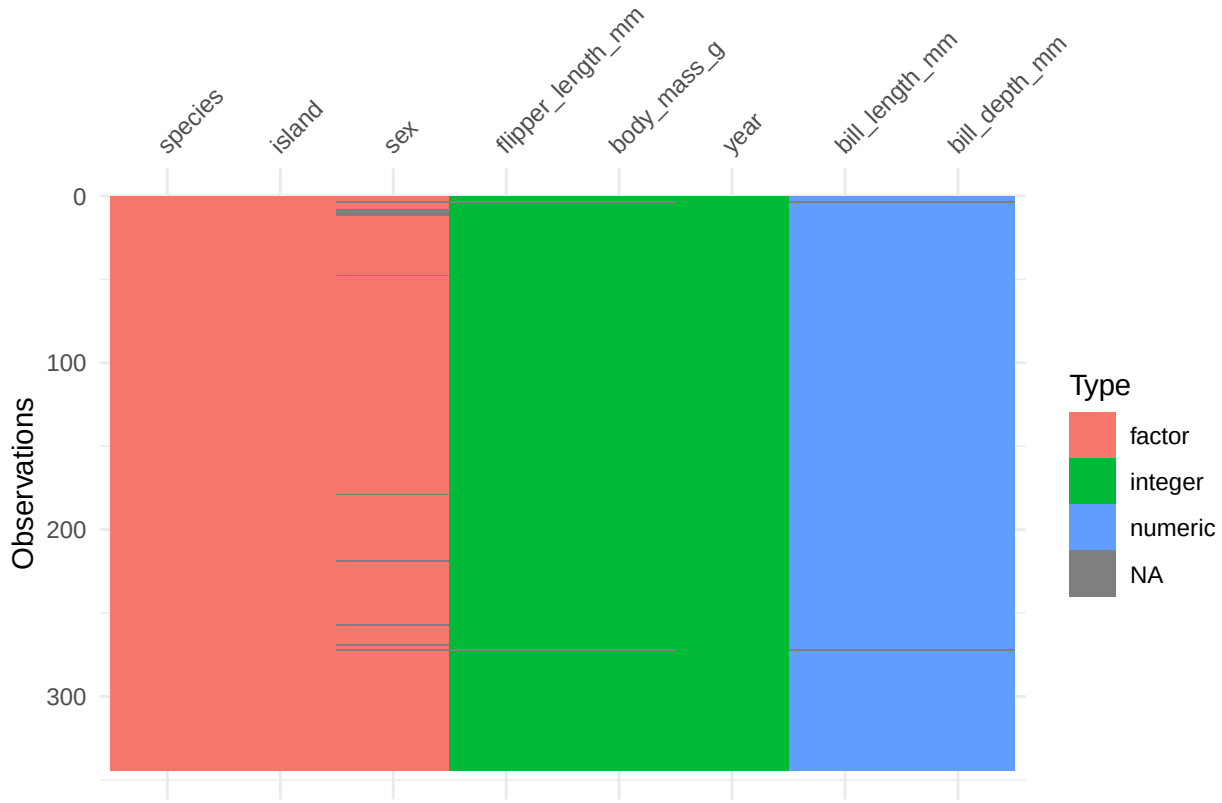
```
dim(ds)
```

```
## [1] 344   8
```

```
glimpse(ds)
```

```
## Rows: 344
## Columns: 8
## $ species           <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, A...
## $ island            <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torge...
## $ bill_length_mm    <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.9, 39.2, 34....
## $ bill_depth_mm     <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20.6, 17.8, 19.6, 18....
## $ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 181, 195, 193, 190, ...
## $ body_mass_g       <int> 3750, 3800, 3250, NA, 3450, 3650, 3625, 4675, 347...
## $ sex               <fct> male, female, female, NA, female, male, female, m...
## $ year              <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2...
```

```
visdat::vis_dat(ds)
```



LDA: Linear Discrimant Analysis

http://www.sthda.com/english/articles/36-classification-methods-essentials/146-discriminant-analysis-essentials-in-r/

```
# Load the data
data("iris")

#Split the data into training (80%) and test set (20%)
set.seed(123)
training.samples <- iris$Species %>%
  createDataPartition(p = 0.8, list=FALSE)
train.data <- iris[training.samples, ]
test.data <- iris[-training.samples, ]

#2. Normalize the data. Categorial variables are automatically ignored.
# Estimate preprocessing parameters
preproc.param <- train.data %>%
  preProcess(method = c("center", "scale"))

# Transform the data using the estimated parameters
train.transformed <- preproc.param %>% predict(train.data)
test.transformed <- preproc.param %>% predict(test.data)
```

```
# Fit the model
model <- lda(Species~., data = train.transformed)
# Make predictions
predictions <- model %>% predict(test.transformed)
# Model accuracy
mean(predictions$class == test.transformed$Species)
```

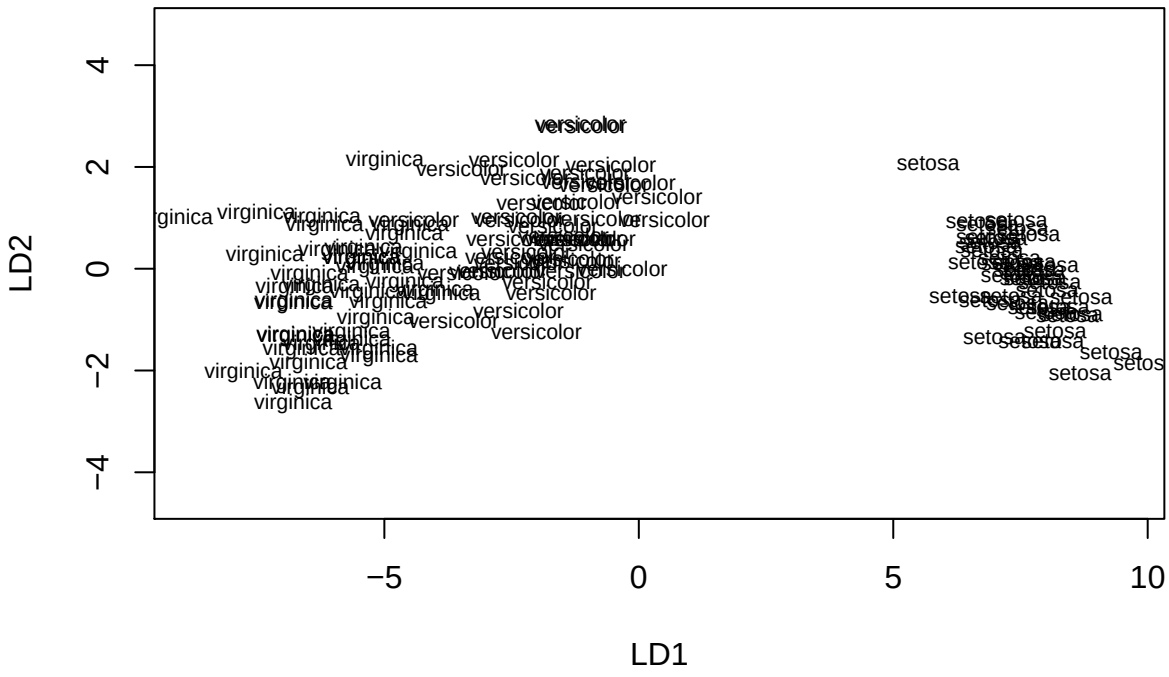
```
## [1] 0.9666667
```

```
# Output Model
model
```

```
## Call:
## lda(Species ~ ., data = train.transformed)
##
## Prior probabilities of groups:
##     setosa versicolor  virginica
##  0.3333333  0.3333333  0.3333333
##
## Group means:
##            Sepal.Length Sepal.Width Petal.Length Petal.Width
## setosa       -1.0112835  0.78048647   -1.2900001  -1.2453195
## versicolor    0.1014181 -0.68674658    0.2566029   0.1472614
## virginica     0.9098654 -0.09373989    1.0333972   1.0980581
##
## Coefficients of linear discriminants:
##                     LD1         LD2
## Sepal.Length  0.6794973  0.04463786
```

```
## Sepal.Width   0.6565085 -1.00330120
## Petal.Length -3.8365047  1.44176147
## Petal.Width  -2.2722313 -1.96516251
##
## Proportion of trace:
##    LD1    LD2
## 0.9902 0.0098
```

```r
# Display model
plot(model)
```




```r
names(predictions)
```

```
## [1] "class"     "posterior" "x"
```

```r
# Predicted classes
head(predictions$class, 6)
```

```
## [1] setosa setosa setosa setosa setosa setosa
## Levels: setosa versicolor virginica
```

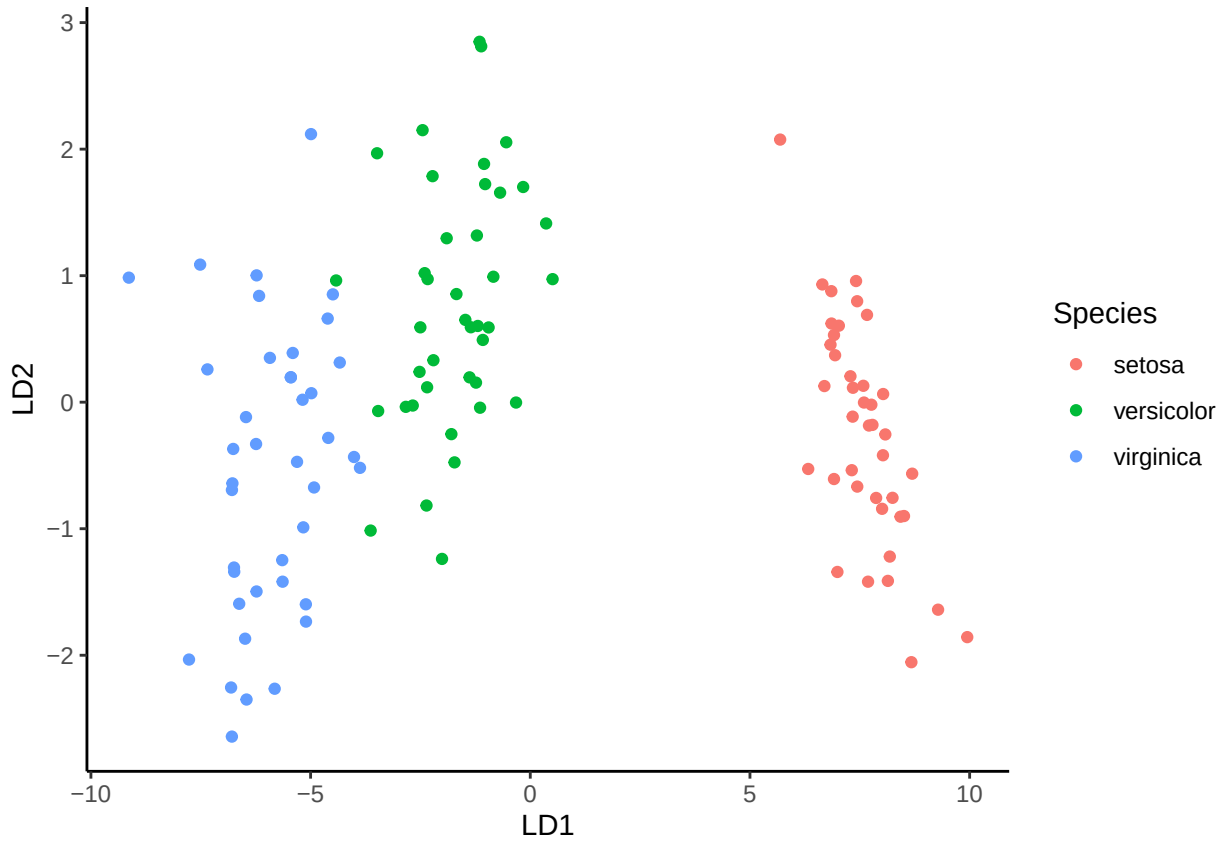
```r
# Predicted probabilities of class membership
head(predictions$posterior, 6)
```

```
##    setosa   versicolor    virginica
## 1       1 3.978425e-22 1.319337e-43
## 2       1 1.038098e-17 3.967605e-38
## 6       1 2.882148e-21 2.041612e-41
## 16      1 8.381782e-28 7.486309e-50
## 23      1 6.531615e-25 3.414098e-47
## 34      1 1.089899e-28 7.865614e-52
```

```r
# Linear discriminants
head(predictions$x, 3)
```

```
##        LD1        LD2
## 1 8.162939 -0.5052768
## 2 7.202713  0.7111062
## 6 7.816243 -1.7327151
```

```r
# Plot
lda.data <- cbind(train.transformed, predict(model)$x)
ggplot(lda.data, aes(LD1, LD2)) +
  geom_point(aes(color = Species))
```




```r
# Model accuracy
mean(predictions$class==test.transformed$Species)
```

```
## [1] 0.9666667
```

```r
sum(predictions$posterior[ ,1] >= .5)
```

```
## [1] 10
```

```r
# QDA

# Fit the model
model <- qda(Species~., data = train.transformed)
model
```

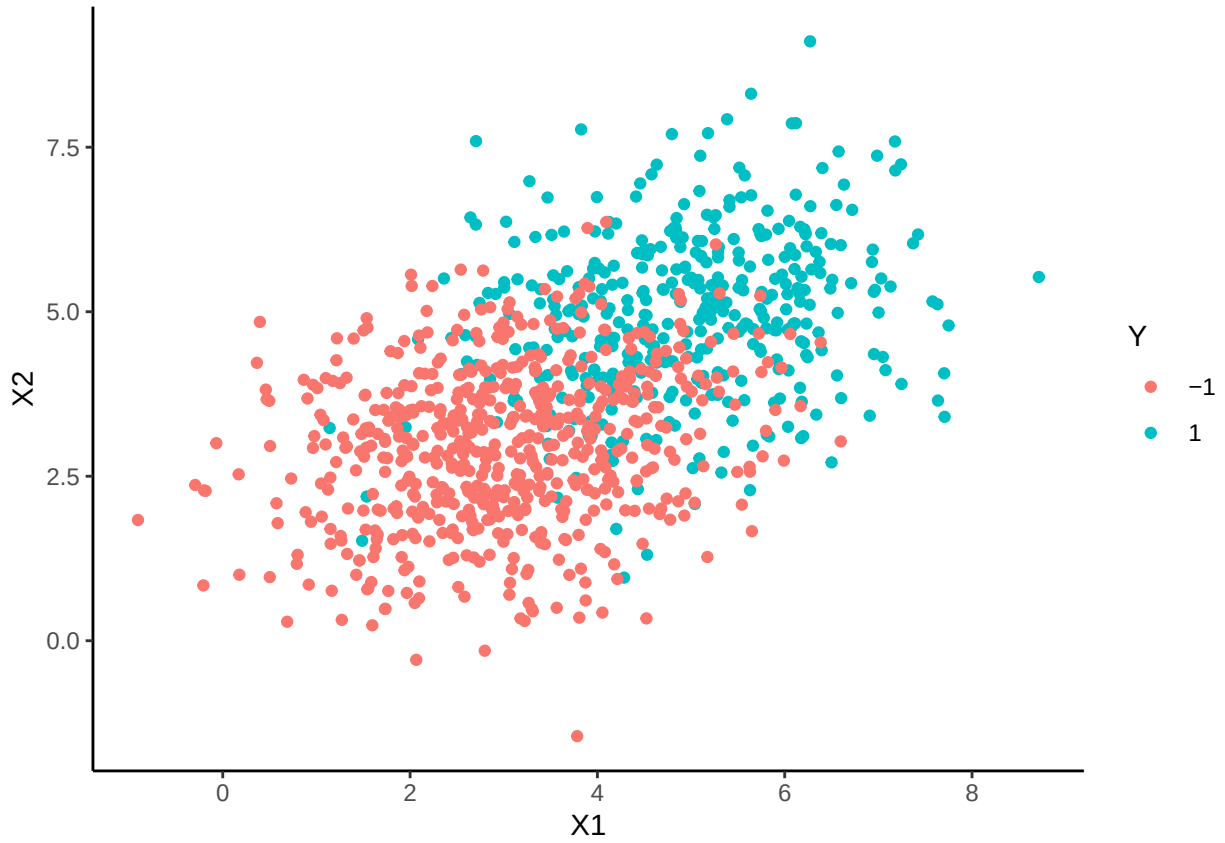
```
## Call:
## qda(Species ~ ., data = train.transformed)
##
## Prior probabilities of groups:
##     setosa versicolor  virginica
##  0.3333333  0.3333333  0.3333333
##
## Group means:
##            Sepal.Length Sepal.Width Petal.Length Petal.Width
## setosa       -1.0112835  0.78048647   -1.2900001  -1.2453195
## versicolor    0.1014181 -0.68674658    0.2566029   0.1472614
## virginica     0.9098654 -0.09373989    1.0333972   1.0980581
```

```r
# Make predictions
predictions <- model %>% predict(test.transformed)
# Model accuracy
mean(predictions$class == test.transformed$Species)
```

```
## [1] 0.9666667
```

https://www.geeksforgeeks.org/linear-discriminant-analysis-in-r-programming/

```r
# Variance Covariance matrix for random bivariate gaussian sample
var_covar <- matrix(data = c(1.5, 0.4, 0.4, 1.5), nrow=2)

# Random bivariate Gaussian samples for class +1
Xplus1 <- rmvnorm(400, mean = c(5, 5), sigma = var_covar)

# Random bivariate Gaussian samples for class -1
Xminus1 <- rmvnorm(600, mean = c(3, 3), sigma = var_covar)

# Samples for the dependent variable
Y_samples <- c(rep(1, 400), rep(-1, 600))

# Combining the independent and dependent variables into a dataframe
dataset <- as.data.frame(cbind(rbind(Xplus1, Xminus1), Y_samples))
colnames(dataset) <- c("X1", "X2", "Y")
dataset$Y <- as.character(dataset$Y)

# Plot the above samples and color by class labels
ggplot(data = dataset) + geom_point(aes(X1, X2, color = Y))
```



QDA: Quadratic Discrimant Analysis

Same link as above

NB: Naive Bayes

https://www.r-bloggers.com/2018/01/understanding-naive-bayes-classifier-using-r/

```
library(e1071)

# Next load the Titantic dataset
data("Titanic")

# Save into a data frame and view it
t_df <- as.data.frame(Titanic)

# Creating data from table
repeating_sequence <- rep.int(seq_len(nrow(t_df)), t_df$Freq)

# Create the dataset by row repetition created
t_ds <- t_df[repeating_sequence, ]
```

```r
# We no longer need the frequency, drop the feature
t_ds$Freq = NULL

# Fitting the Naive Bayes model
nbm <- naiveBayes(Survived~., data=t_ds)
# Output the model
nbm
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##       No      Yes
## 0.676965 0.323035
##
## Conditional probabilities:
##      Class
## Y           1st        2nd        3rd       Crew
##   No  0.08187919 0.11208054 0.35436242 0.45167785
##   Yes 0.28551336 0.16596343 0.25035162 0.29817159
##
##      Sex
## Y          Male     Female
##   No  0.91543624 0.08456376
##   Yes 0.51617440 0.48382560
##
##      Age
## Y          Child      Adult
##   No  0.03489933 0.96510067
##   Yes 0.08016878 0.91983122
```

```r
# Prediction on the dataset
nb_predictions <- predict(nbm, t_ds)
# Confusion matrix to check accuracy
table(nb_predictions, t_ds$Survived)
```

```
##
## nb_predictions   No  Yes
##            No  1364  362
##            Yes  126  349
```

```r
# Getting started with Naive Bayes in mlr
library(mlr)
```

```
## Loading required package: ParamHelpers
```

```
## 'mlr' is in maintenance mode since July 2019. Future development
## efforts will go into its successor 'mlr3' (<https://mlr3.mlr-org.com>).
```

```
##
## Attaching package: 'mlr'

## The following object is masked from 'package:e1071':
##
##     impute

## The following object is masked from 'package:caret':
##
##     train
```

```
# Create a classification task for learning on Titantic Dataset and specify the target feature
task <- makeClassifTask(data = t_ds, target="Survived")

# Initialize the Naive Bayes classifier
selected_model <- makeLearner("classif.naiveBayes")

# Train the model
nb_mlr <- train(selected_model, task)

# Read the model learned
nb_mlr$learner.model
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##       No      Yes
## 0.676965 0.323035
##
## Conditional probabilities:
##      Class
## Y            1st        2nd        3rd       Crew
##   No  0.08187919 0.11208054 0.35436242 0.45167785
##   Yes 0.28551336 0.16596343 0.25035162 0.29817159
##
##      Sex
## Y           Male     Female
##   No  0.91543624 0.08456376
##   Yes 0.51617440 0.48382560
##
##      Age
## Y          Child      Adult
##   No  0.03489933 0.96510067
##   Yes 0.08016878 0.91983122
```

```
# Predict on the dataset without passing the target feature
predictions_mlr <- as.data.frame(predict(nb_mlr, newdata = t_ds[,1:3]))
```

```r
# Confusion matrix to check accuracy
table(predictions_mlr[,1], t_ds$Survived)
```

```
##
##         No  Yes
##   No  1364  362
##   Yes  126  349
```

https://www.geeksforgeeks.org/naive-bayes-classifier-in-r-programming/

==== Prompt =====