HarvardX PH125.9x
Data Science: Capstone
Penguins Project

Tom Hannigan
January 7, 2021

Introduction

This is the second of two capstone projects for completion of the ninth and final course in HarvardX's multi-part Data Science Professional Certificate series. The scope of this project is to utilize various physical measurements of penguins as a way to correctly identify their species. The data is provided from the “palmerpenguins” package. The accuracy of the predict function will be our metric for this project.

The goal of the project is to generate and train a machine learning algorithm using physical measurements identified in the “data_train” partition to predict species in the “data_test” dataset. When the machine learning is evaluated on the “data_test” partition, the goal is to generate an **Accuracy > 98%** on the overall accuracy from the predict function. The calculation to obtain the accuracy is:

```
confusionMatrix(predict(fit_model, data_test), data_test$species) $overall["Accuracy"]
```

The steps performed for the project were:

- Download the dataset and review
- Scrub and simplify the dataset
- Partition the dataset
- Visually evaluate the data
- Identify and model various fit models
- Determine and compare the accuracies on the “data_test” partition.
- Determine and report the optimal method and accuracy

Methods/Analysis

The data was pulled from the penguins data set. A summary for review was generated with the following code:

----- CODE ----

```
install.packages("palmerpenguins")
library(palmerpenguins)
data(package = 'palmerpenguins')
summary(penguins)
```

----- RESULT ----

```
> summary(penguins)
      species          island    bill_length_mm  bill_depth_mm   flipper_length_mm  body_mass_g       sex           year
 Adelie   :152   Biscoe   :168   Min.   :32.10   Min.   :13.10   Min.   :172.0     Min.   :2700   female:165   Min.   :2007
 Chinstrap: 68   Dream    :124   1st Qu.:39.23   1st Qu.:15.60   1st Qu.:190.0     1st Qu.:3550   male  :168   1st Qu.:2007
 Gentoo   :124   Torgersen: 52   Median :44.45   Median :17.30   Median :197.0     Median :4050   NA's  : 11   Median :2008
                                 Mean   :43.92   Mean   :17.15   Mean   :200.9     Mean   :4202                Mean   :2008
                                 3rd Qu.:48.50   3rd Qu.:18.70   3rd Qu.:213.0     3rd Qu.:4750                3rd Qu.:2009
                                 Max.   :59.60   Max.   :21.50   Max.   :231.0     Max.   :6300                Max.   :2009
                                 NA's   :2       NA's   :2       NA's   :2         NA's   :2
```

In reviewing the summary it is identified that there are several items that need to be addressed with the dataset. These are the tasks that are performed on the data to make it simplified for this project

- Remove the entries that include an NA in any of the variable columns
- Remove columns that are not pertinent to the project to simplify the dataset
- Rename some columns for better presentation
- Confirm the results

----- CODE ----

```
wpenguins <- penguins[!is.na(penguins$bill_length_mm) | !is.na(penguins$bill_depth_mm) | !is.na(penguins$flipper_length_mm) | !is.na(penguins$body_mass_g),]

simplep <- wpenguins %>% select(length=bill_length_mm, depth=bill_depth_mm, flipper=flipper_length_mm, body=body_mass_g, species)

summary(!is.na(simplep))

head(simplep)
```

----- RESULT ----

```
> summary(!is.na(simplep))
  length          depth          flipper           body          species
 Mode:logical   Mode:logical   Mode:logical   Mode:logical   Mode:logical
 TRUE:342       TRUE:342       TRUE:342       TRUE:342       TRUE:342
> head(simplep)
  length depth flipper body species
1   39.1  18.7     181 3750  Adelie
2   39.5  17.4     186 3800  Adelie
3   40.3  18.0     195 3250  Adelie
5   36.7  19.3     193 3450  Adelie
6   39.3  20.6     190 3650  Adelie
7   38.9  17.8     181 3625  Adelie
```

The remaining dataset needs to be partitioned to set up a training dataset and a test dataset. It will be split at a 20% partition. The following code was used to accomplish that:

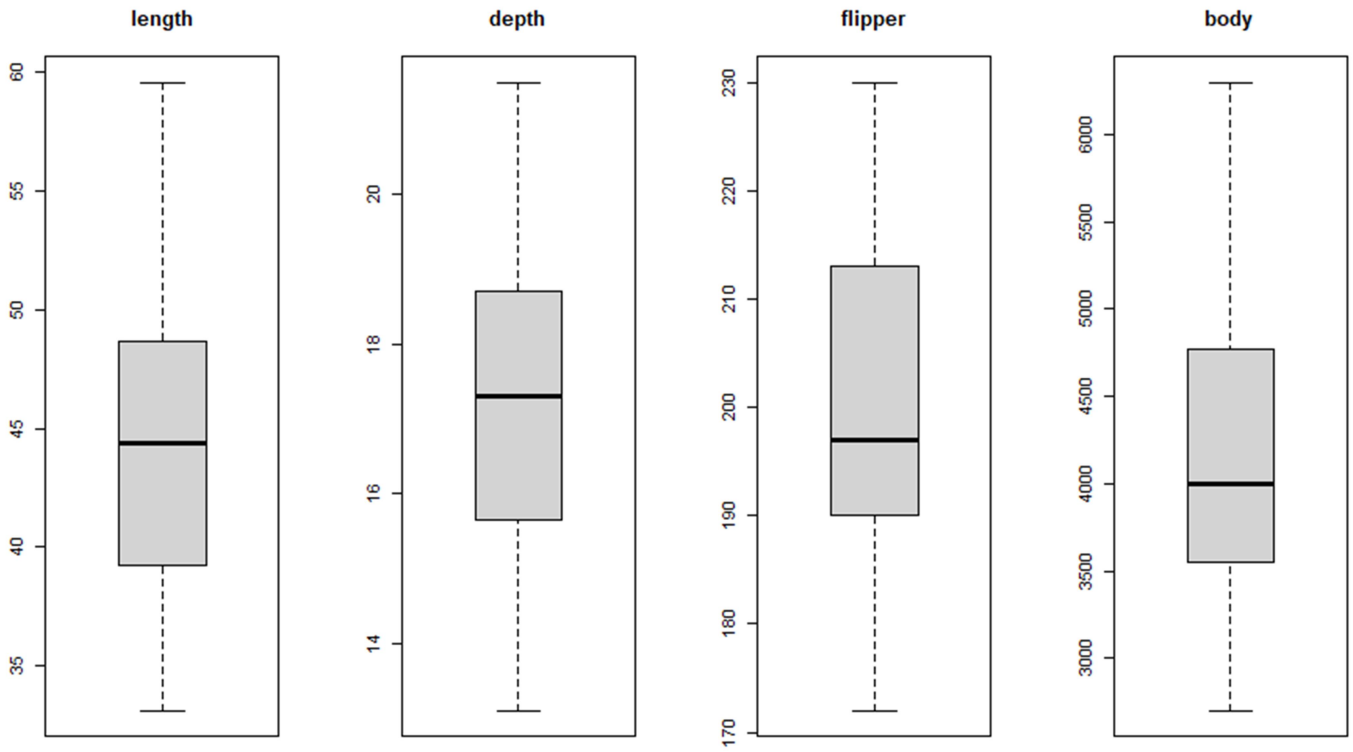
----- CODE ----

```
set.seed(68, sample.kind="Rounding")
trainIndex <- createDataPartition(simplep$species, p=0.8, list=FALSE)
data_train <- simplep[trainIndex,]
data_test <- simplep[-trainIndex,]
```

----- RESULT ----

```
> summary(data_train)
     length          depth          flipper          body          species
 Min.   :33.10   Min.   :13.10   Min.   :172   Min.   :2700   Adelie   :121
 1st Qu.:39.20   1st Qu.:15.65   1st Qu.:190   1st Qu.:3550   Chinstrap: 55
 Median :44.40   Median :17.30   Median :197   Median :4000   Gentoo   : 99
 Mean   :43.92   Mean   :17.15   Mean   :201   Mean   :4198
 3rd Qu.:48.65   3rd Qu.:18.70   3rd Qu.:213   3rd Qu.:4775
 Max.   :59.60   Max.   :21.50   Max.   :230   Max.   :6300
> summary(data_test)
     length          depth          flipper          body          species
 Min.   :32.10   Min.   :13.20   Min.   :180.0   Min.   :3000   Adelie   :30
 1st Qu.:39.70   1st Qu.:15.40   1st Qu.:190.0   1st Qu.:3625   Chinstrap:13
 Median :44.50   Median :17.30   Median :195.0   Median :4200   Gentoo   :24
 Mean   :43.93   Mean   :17.14   Mean   :200.5   Mean   :4219
 3rd Qu.:47.45   3rd Qu.:18.65   3rd Qu.:214.0   3rd Qu.:4750
 Max.   :58.00   Max.   :20.70   Max.   :231.0   Max.   :5950
```

Now we will visually evaluate the train data. The first plot is a box plot of each of the predictors. In looking at it, this shows that we have variation in each of the predictors, but really doesn't give us a lot of insight related to the species:

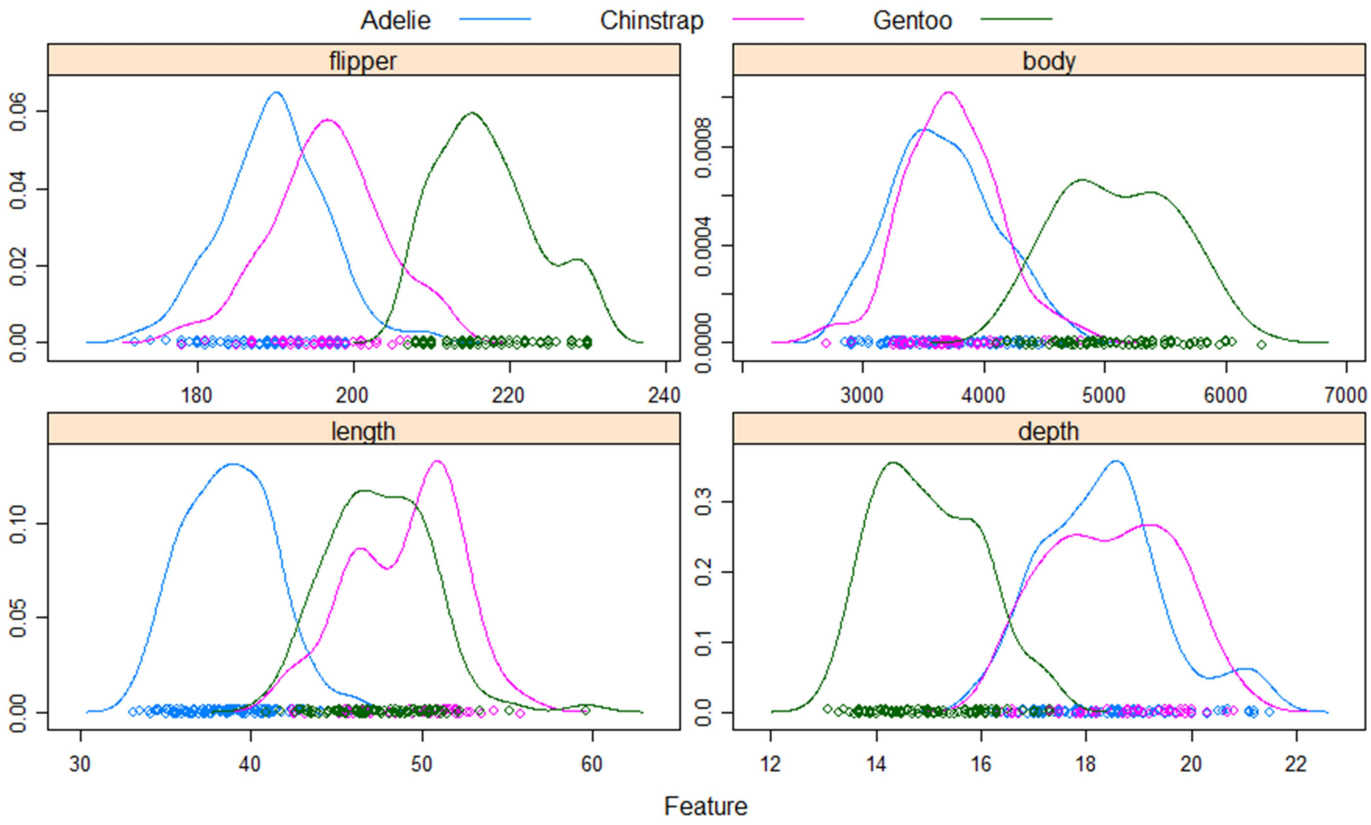


----- CODE ----

```
x <- data_train[, 1:4]
y <- factor(data_train[,5])

par(mfrow=c(1,4))
for(i in 1:4) {
  boxplot(x[,i], main=names(x)[i])
}
```

The next plot to generate is a density plot of the predictors. In looking at the plot, you can see areas where the values for predictor are skewed for one or more species in relation to the other species. For example, there is almost no overlap between Adelie and Gentoo species for the flipper length.

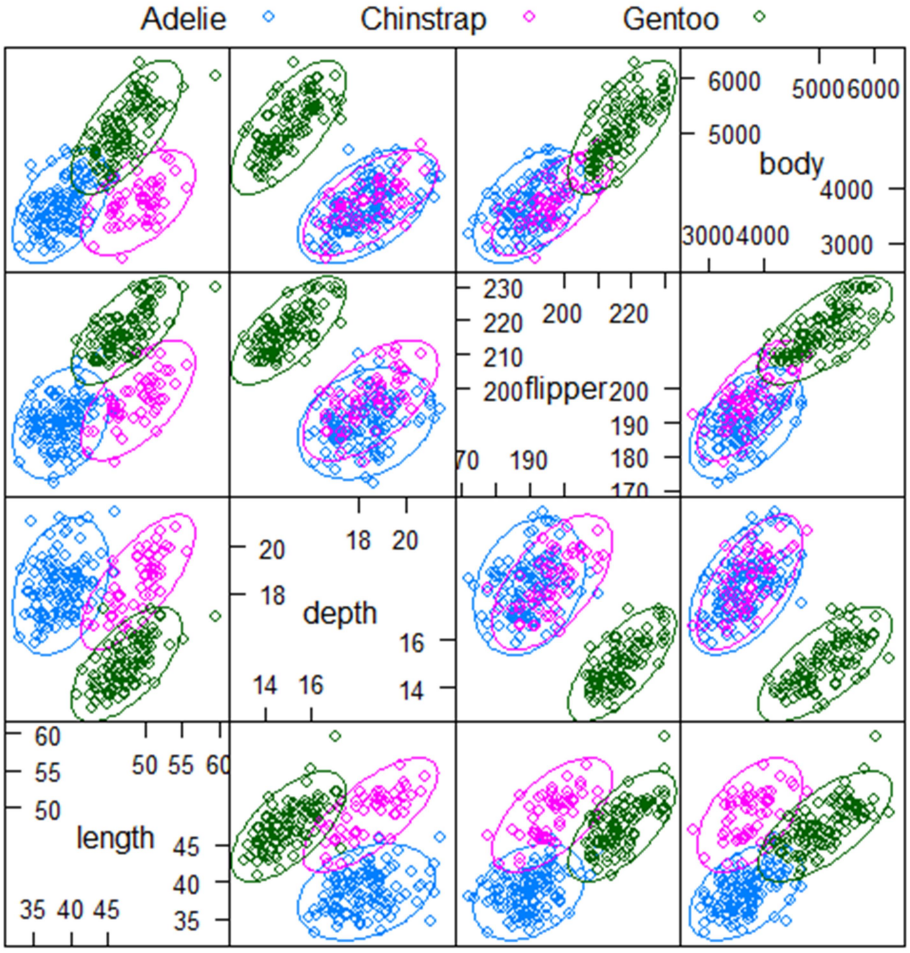


----- CODE ----

```
x <- data_train[, 1:4]
y <- factor(data_train[,5])

scales <- list(x=list(relation="free"), y=list(relation="free"))
featurePlot(x=x, y=y, plot="density", scales=scales, auto.key = list(columns = 3))
```

The final plot to evaluate is a scatter plot matrix. It is very clear that there are several data combinations that would be good indicators. A prime example is the clear distinction between the bill depth and the body weight for Gentoo species versus the other two. However, it would be completely ineffective to distinguish between the Adelie and Chinstrap species. Although not as definitive as the depth\body is for the Gentoo, the bill length and bill depth interaction between Adelie and Chinstrap shows a potential to distinguish between those two species.



Scatter Plot Matrix

----- CODE ----

```
x <- data_train[, 1:4]
y <- factor(data_train[,5])

featurePlot(x, y, plot = "ellipse", auto.key = list(columns = 3))
```

The next step will be to generate several fit models (utilizing different methods in r) and evaluate their performance. First will be the classification and regression trees or CART model. This is generated with the code/formulas below:

----- CODE ----

```
set.seed(68, sample.kind="Rounding")
cart_fit <- train(species~., data=data_train, method="rpart")
```

----- RESULT ----

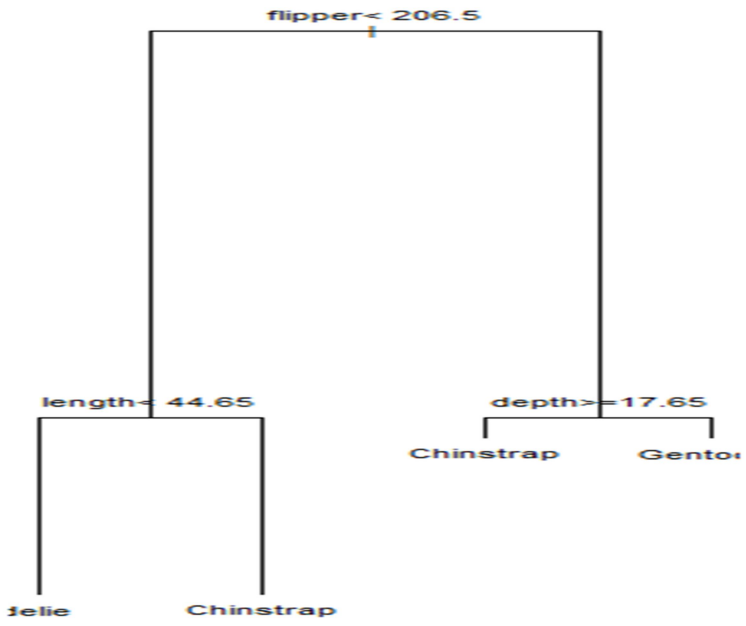
```
CART

275 samples
  4 predictor
  3 classes: 'Adelie', 'Chinstrap', 'Gentoo'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 275, 275, 275, 275, 275, 275, ...
Resampling results across tuning parameters:

  cp          Accuracy   Kappa
  0.03246753  0.9376097  0.9013253
  0.28571429  0.8966916  0.8307487
  0.62987013  0.5113544  0.1173598

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.03246753.
```



The next one will be Random Forest. This is intended to improve prediction performance and reduce instability by averaging multiple decision trees. This is generated with the code/formulas below:

----- CODE ----

```
set.seed(68, sample.kind="Rounding")
rf_fit <- randomForest(species ~ ., data=data_train)
```

----- RESULT ----

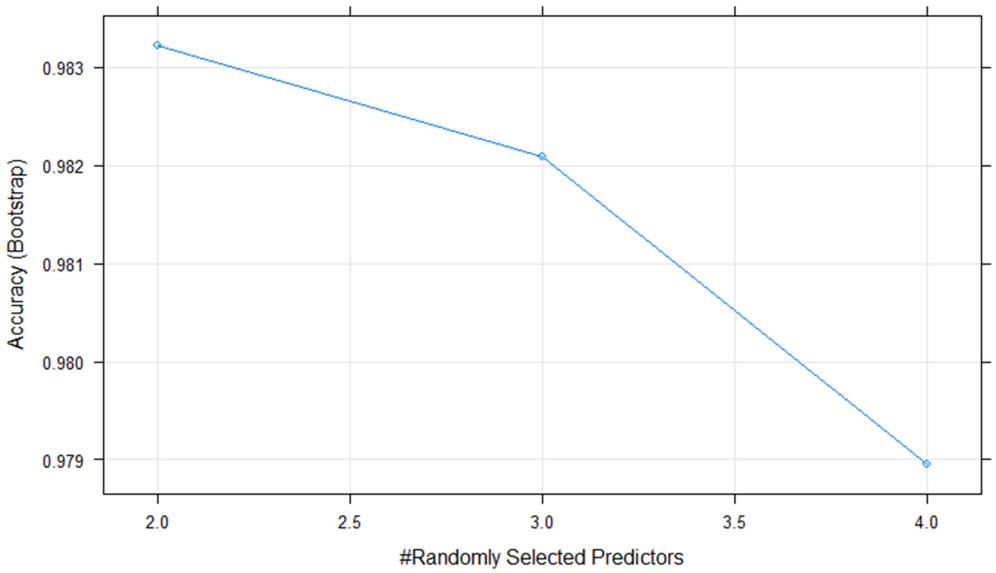
```
Random Forest

275 samples
  4 predictor
  3 classes: 'Adelie', 'Chinstrap', 'Gentoo'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 275, 275, 275, 275, 275, 275, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
  2     0.9832233  0.9733178
  3     0.9820895  0.9715283
  4     0.9789511  0.9666203

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.
```



The next one will be linear discriminant analysis or LDA model. This is designed to find a linear combination of predictors that separates the class objects, in this case the species. This is generated with the code/formulas below:

----- CODE ----

```
set.seed(68, sample.kind="Rounding")
lda_fit <- train(species~., data=data_train, method="lda")
```

----- RESULT ----

```
Linear Discriminant Analysis

275 samples
  4 predictor
  3 classes: 'Adelie', 'Chinstrap', 'Gentoo'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 275, 275, 275, 275, 275, 275, ...
Resampling results:

  Accuracy   Kappa
  0.9877245  0.9804423
```

The next one is the quadratic discriminant analysis or QDA model. This similar to LDA and is a version of Naïve Bayes where the distributions are assumed multivariate normal. This is generated with the code/formulas below:

----- CODE ----

```
set.seed(68, sample.kind="Rounding")
qda_fit <- train(species~., data=data_train, method="qda")
```

----- RESULT ----

```
Quadratic Discriminant Analysis

275 samples
  4 predictor
  3 classes: 'Adelie', 'Chinstrap', 'Gentoo'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 275, 275, 275, 275, 275, 275, ...
Resampling results:

  Accuracy   Kappa
  0.9852935  0.9765758
```

The last one we will be evaluating is the support vector machine or SVM model. This tries to determine the hyper-plane to separate the classes in multiple dimensions. We will do two kernel variations, “radial” and “polynomial” for the SVM model. These are generated with the code/formulas below:

----- CODE ----

```
# With radial kernel
set.seed(68, sample.kind="Rounding")
svm_fit_r <- svm(species~., data=data_train, kernel='radial')
```

----- RESULT ----

```
Call:
svm(formula = species ~ ., data = data_train, kernel = "radial")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  48
```

----- CODE ----

```
# With polynomial kernel
set.seed(68, sample.kind="Rounding")
svm_fit_p <- svm(species~., data=data_train, kernel='polynomial')
```

----- RESULT ----

```
Call:
svm(formula = species ~ ., data = data_train, kernel = "polynomial")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  polynomial
       cost:  1
     degree:  3
     coef.0:  0

Number of Support Vectors:  72
```

Results

We will now take all of the various trained models; CART, RF, LDA, QDA, SVM_R and SVM_P and predict them against the test dataset. Recall that our goal is an accuracy > 98%. These are generated with the code/formulas below:

----- CODE ----

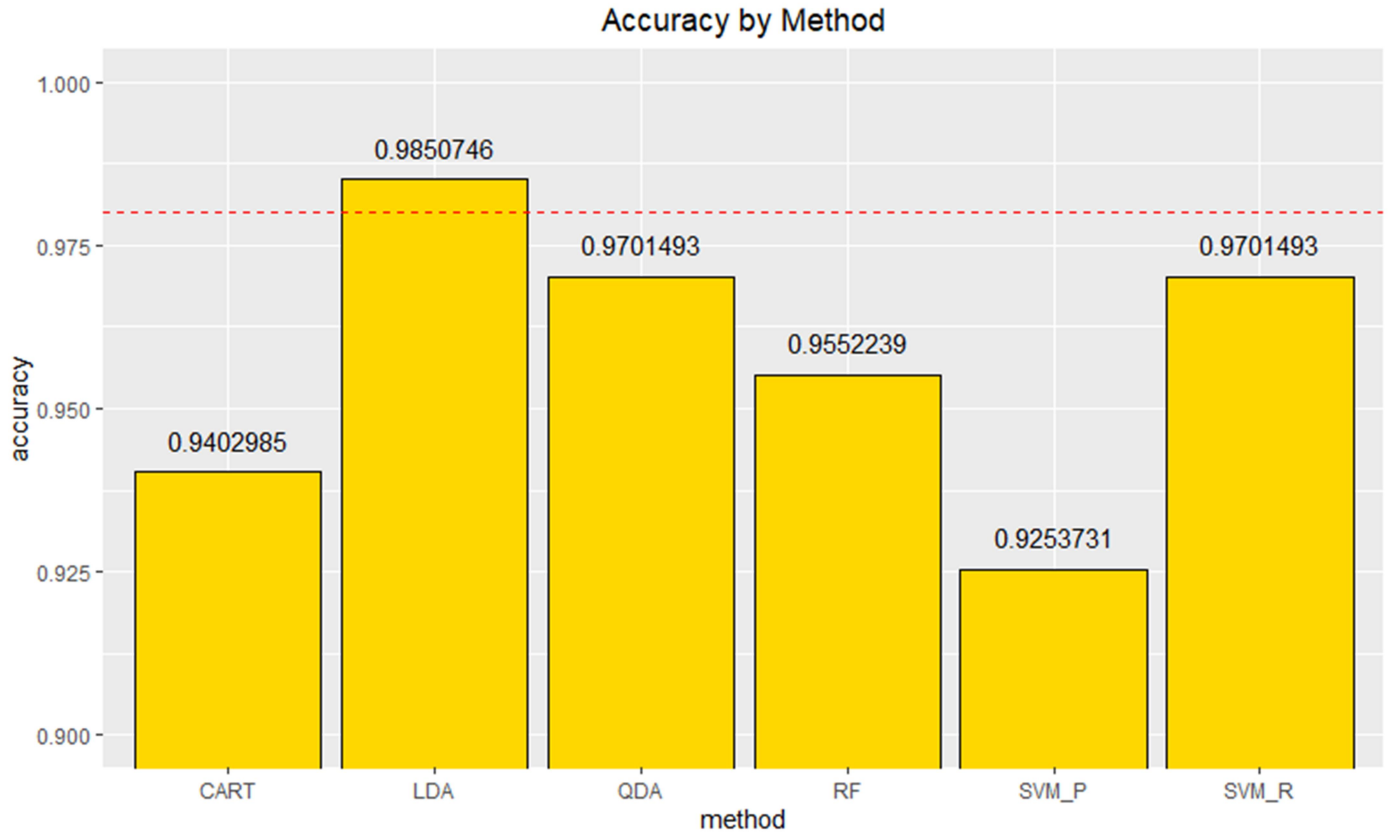
```
cart_acc <- confusionMatrix(predict(cart_fit, data_test), data_test$species)$overall["Accuracy"]
rf_acc <- confusionMatrix(predict(rf_fit, data_test), data_test$species)$overall["Accuracy"]
lda_acc <- confusionMatrix(predict(lda_fit, data_test), data_test$species)$overall["Accuracy"]
qda_acc <- confusionMatrix(predict(qda_fit, data_test), data_test$species)$overall["Accuracy"]
svm_acc_r <- confusionMatrix(predict(svm_fit_r, data_test), data_test$species)$overall["Accuracy"]
svm_acc_p <- confusionMatrix(predict(svm_fit_p, data_test), data_test$species)$overall["Accuracy"]
```

----- RESULT ----

method	accuracy
CART	0.9402985
RF	0.9552239
LDA	0.9850746
QDA	0.9701493
SVM_R	0.9701493
SVM_P	0.9253731

Conclusion

We were able to exceed the project goal of an accuracy > 98% utilizing the LDA fit model. Its predicted accuracy on the test dataset was 98.51%. This is graphed below:



----- CODE ----

```
acc_results %>% ggplot(aes(method,accuracy)) +
  geom_col(color = "black", fill = "gold") +
  geom_text(aes(label = format(accuracy, digits = 7)), nudge_y = 0.005) +
  coord_cartesian(ylim=c(0.9, 1.0)) +
  geom_abline(slope=0, intercept=0.98,  col = "red",lty=2) +
  ggtitle("Accuracy by Method") +
  theme(plot.title = element_text(hjust = 0.5))
```

This met the stated goal, but there is opportunity to improve the results. The analysis was only performed on the variables as predictors for the species. There were other attribute data columns such as island and sex that were not factored into the evaluation.

A limitation of this project was the length and width of the dataset. There were only 4 variables evaluated for 342 penguins. There are many more measurements that could be done on the penguins, such as height, foot length, etc. That additional data could be utilized to further improve the models. Additionally, you would expect that modeling off of a larger dataset would be expected to improve the accuracy.

In order to continue the project and strive towards 100 percent accuracy, we would want to expand the analysis to include many other factors, both variable and attribute. Additionally, the dataset included several NA values. This could indicate difficulty in obtaining these measurements. It would be ideal to investigate details and measurements that could be obtained remotely with the use of technology. Activities such as facial and marking recognition, laser measurements, such as for height, obtained from a distance or photograph, radar/laser evaluations of speed, etc., could be the next step in improving the accuracy and feasibility of predicting a large waddle of penguins.

Citations:

HarvardX PH125.9x Capstone Course: https://www.edx.org/professional-certificate/harvardx-data-science

Textbook: "Introduction to Data Science", by Rafael A. Irizarry: https://rafalab.github.io/dsbook/

palmerpenguins:

#> Horst AM, Hill AP, Gorman KB (2020). palmerpenguins: Palmer
#> Archipelago (Antarctica) penguin data. R package version 0.1.0.
#> https://allisonhorst.github.io/palmerpenguins/. doi:
#> 10.5281/zenodo.3960218.