Determining the gender of a penguin: An Exploratory and Classification Analysis of The Penguin Dataset

GIFT E. IDAMA
Computer And Information Sciences Department
Towson University,
Towson, MD, USA.
gidama1@students.towson.edu

Abstract: The idea of the classification algorithm is very simple. We predict the target class by analyzing the training dataset. We use training datasets to obtain better boundary conditions that can be used to determine each target class. Once the boundary condition is determined, the next task is to predict the target class. The entire process is known as classification. Classification is a form of supervised learning where the response variable is categorical, as opposed to numeric for regression. ***Our goal is to find a rule, algorithm, or function which takes as input a feature vector, and outputs a category which is the true category as often as possible***.

Keywords: penguin, variable, classification, Exploratory analysis, training, testing, validation, confusion Matrix.

1 INTRODUCTION

What is classification: Classification is the process of predicting a categorical label of a data object based on its features and properties. In classification, we locate identifiers or boundary conditions that correspond to a label or category.

Generally, classifiers in R are used to predict specific category related information like reviews or ratings such as good, best or worst. Some of the various classifiers out there are decision trees, Naïve Bayes classifiers, knn classifiers, support vector machines SVM etc.

Caret package: A set of functions that attempt to streamline the process for creating predictive models. The package contains tools for data splitting, preprocessing, feature selection, model tuning, using resampling, variable importance estimation.

2 PROJECT OBJECTIVES

This project looks to make analysis on the penguin dataset and make predictions using classification methods. The specific goals of this project are.

1. Perform Exploratory analysis on the dataset to understand the features of the dataset, the variables and the relationship between each variable.
2. To partition our dataset into 2 parts, training and test partition
3. To build a classification model using various classification methods and to fit our train data into this model in order to train our data to make predictions.
4. To predict the gender of a penguin based on its physical characteristics.
5. To validate our model's predictions by using a confusion Matrix
6. To compare each classification algorithm used based on the accuracy metric to see model performed best and which method is more accurate in making predictions.
7. To determine which variable played a very important role in determining the gender of the penguin.

3 DESCRIPTIONS OF THE PENGUIN DATASET:

The dataset contain data for 344 penguins. There are 3 different species of penguins in the dataset collected from 3 islands in the palmer archipelago, Antarctica. The species include Adelie, Chinstrap, and Gentoo. It also shows the island on which these species can be found. The 3 islands in our dataset are Torgersen, Biscoe and Dream Island. There are about 344 observations, and 8 variables present in our dataset. It also contains the physical characteristics of each specie as well as the genders of the penguins.

The variables include specie, island, bill length, bill depth, flipper length, body mass, sex and year. 3

variables are categorical (sex, island, species) while the rest are numeric.

Penguins Data Column Definition

- **species** a factor denoting penguin species (Adelie, Chinstrap and Gentoo)
- **island** a factor denoting island in Palmer Archipelago, Antarctica (Biscoe, Dream or Torgersen)
- **bill_length_mm** a number denoting bill length (millimeters)
- **bill_depth_mm** a number denoting bill depth (millimeters)
- **flipper_length_mm** an integer denoting flipper length (millimeters)
- **body_mass_g** an integer denoting body mass (grams)
- **sex** a factor denoting penguin sex (female, male)

```
> str(penguins)
tibble [344 x 8] (S3: tbl_df/tbl/data.frame)
 $ species          : Factor w/ 3 levels "Adelie","Chinstrap",..: 1 1 1 1 1 1 1 1
 $ island           : Factor w/ 3 levels "Biscoe","Dream",..: 3 3 3 3 3 3 3 3 3 3
 $ bill_length_mm   : num [1:344] 39.1 39.5 40.3 NA 36.7 39.3 38.9 39.2 34.1 42 ..
 $ bill_depth_mm    : num [1:344] 18.7 17.4 18 NA 19.3 20.6 17.8 19.6 18.1 20.2 ..
 $ flipper_length_mm: int [1:344] 181 186 195 NA 193 190 181 195 193 190 ...
 $ body_mass_g      : int [1:344] 3750 3800 3250 NA 3450 3650 3625 4675 3475 4250
 $ sex              : Factor w/ 2 levels "female","male": 2 1 1 NA 1 2 1 2 NA NA ..
 $ year             : int [1:344] 2007 2007 2007 2007 2007 2007 2007 2007 2007 20(
>
```

***Fig 1**: Structure of the data set*

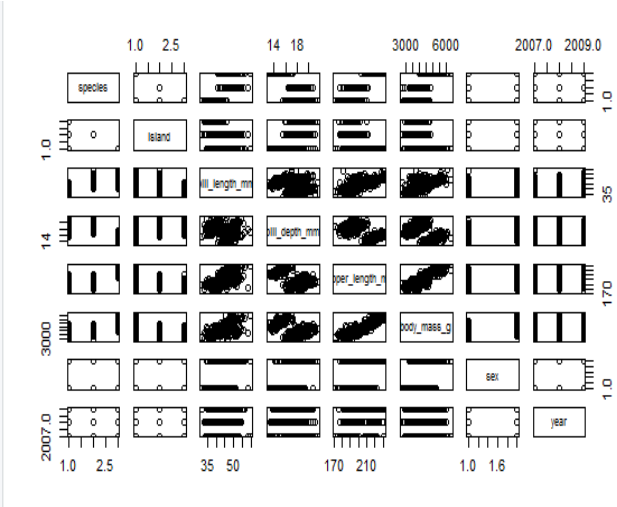

***Fig 2**: Scatterplot of the data set*

4 METHODOLOGIES

Under this section, there are five main steps to carry out experiment on our dataset,

1. Explore our data.
2. Divide our data set into partitions.
3. Train our model.
4. Evaluate model.
5. Compare model results.

4.1 Exploratory Data Analysis:

Before we dive into data analysis, we must do some data wrangling in order to deal with the missing values. Let us view our data set and check how many missing values we have.

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
1	Adelie	Torgersen	39.1	18.7	181	3750	male
2	Adelie	Torgersen	39.5	17.4	186	3800	female
3	Adelie	Torgersen	40.3	18.0	195	3250	female
4	Adelie	Torgersen	*NA*	*NA*	*NA*	*NA*	*NA*
5	Adelie	Torgersen	36.7	19.3	193	3450	female
6	Adelie	Torgersen	39.3	20.6	190	3650	male
7	Adelie	Torgersen	38.9	17.8	181	3625	female
8	Adelie	Torgersen	39.2	19.6	195	4675	male
9	Adelie	Torgersen	34.1	18.1	193	3475	*NA*
10	Adelie	Torgersen	42.0	20.2	190	4250	*NA*
11	Adelie	Torgersen	37.8	17.1	186	3300	*NA*

ing 1 to 12 of 344 entries, 8 total columns

***Fig 3**: View of the dataset*

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

***Fig 4**: Summary of the data set*

Taking the summary of the dataset, we can see that some missing values exists in the dataset. 2 missing values present in the bill_length_mm, bill_depth_mm, flipper_length_mm, body_mass_g columns, while there are about 11 missing values in the sex column.

The missing values in each column does not affect our dataset significantly, so eliminating these missing values should be straight forward. We can make use of the **"na.omit() function" to** remove the rows with **NA's.** After calling the function, we arrive at a data frame with no missing values present.

```
> na.omit(df)
# A tibble: 333 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  Torger~          39.1          18.7              181        3750 male
 2 Adelie  Torger~          39.5          17.4              186        3800 fema~
 3 Adelie  Torger~          40.3          18                195        3250 fema~
 4 Adelie  Torger~          36.7          19.3              193        3450 fema~
 5 Adelie  Torger~          39.3          20.6              190        3650 male
 6 Adelie  Torger~          38.9          17.8              181        3625 fema~
 7 Adelie  Torger~          39.2          19.6              195        4675 male
 8 Adelie  Torger~          41.1          17.6              182        3200 fema~
 9 Adelie  Torger~          38.6          21.2              191        3800 male
10 Adelie  Torger~          34.6          21.1              198        4400 male
# ... with 323 more rows
```

Fig 5*: Result of calling the na.omit function*

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
1	Adelie	Torgersen	39.1	18.7	181	3750	male
2	Adelie	Torgersen	39.5	17.4	186	3800	female
3	Adelie	Torgersen	40.3	18.0	195	3250	female
4	Adelie	Torgersen	36.7	19.3	193	3450	female
5	Adelie	Torgersen	39.3	20.6	190	3650	male
6	Adelie	Torgersen	38.9	17.8	181	3625	female
7	Adelie	Torgersen	39.2	19.6	195	4675	male
8	Adelie	Torgersen	41.1	17.6	182	3200	female
9	Adelie	Torgersen	38.6	21.2	191	3800	male
0	Adelie	Torgersen	34.6	21.1	198	4400	male

ing 1 to 11 of 333 entries, 8 total columns

Fig 6*: Updated data frame with no missing values*

NOTE: Notice the reduced observation in Figure 6, shows the rows with missing values have been deleted, hence the reduced number of entries.

4.2 Virtual Data Analysis:

First, let do a breakdown of some important factors before performing the graphical representation of our variables.

4.2.2 Number of Species:

There are 146 Adelie species of penguins, 68 Chinstrap species and 119 Gentoo species.

```
> df %>%
+ count(species)
# A tibble: 3 x 2
  species       n
  <fct>     <int>
1 Adelie      146
2 Chinstrap    68
3 Gentoo      119
>
```

Fig 7: *Number of species*

4.2.2 Number of species on each island:

In order to tell how many species are present on each island, we use the count function, with one of the arguments being the island variable, this will give us a breakdown of the species on each island.

```
> count(df, species, island)
# A tibble: 5 x 3
  species   island        n
  <fct>     <fct>     <int>
1 Adelie    Biscoe       44
2 Adelie    Dream        55
3 Adelie    Torgersen    47
4 Chinstrap Dream        68
5 Gentoo    Biscoe      119
>
```

Fig 8: *Number of specie on each island*

So, the Adelie specie can be found on all three islands. (44 in Biscoe, 55 in Dream and 47 in Torgersen). The Chinstrap specie is only found on the Dream Island while Gentoo species are found on the Biscoe Island.

In total, Biscoe Island has 163 penguins on it, Dream Island has 123 penguins on it, while Torgersen has about 47 penguins on it.

```
> df%>%
+ count(island)
# A tibble: 3 x 2
  island        n
  <fct>     <int>
1 Biscoe      163
2 Dream       123
3 Torgersen    47
>
```

Fig 9: *Total number of specie on each island*

Below is a chart showing what island each specie is in, both male and female.

```
> ggplot(df, aes(x=bill_length_mm, flipper_length_mm, color=sex))+
+ geom_point()+
+ facet_grid(island~species)+
+ theme_bw()+
+ ggtitle("penguin bill length and flipper length")
>
```

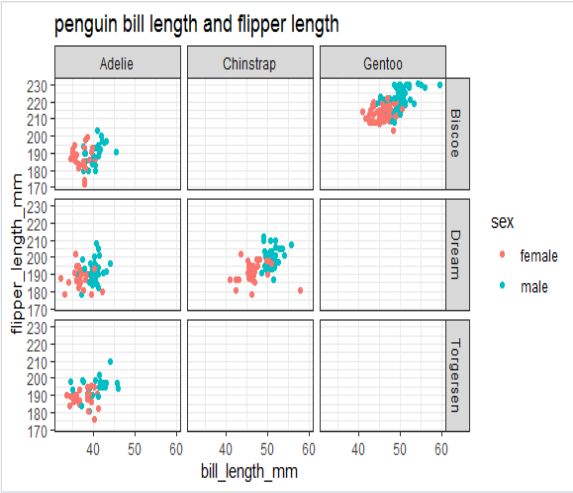

Fig 9: *Chart showing the island each specie is located*

Description of Chart: Adelie is located on all three islands; Chinstrap can be found only on the Dream Island while Gentoo species can be found only on the Biscoe Island.

4. 3 Comparing the Variables:

The next step in our exploratory data analysis is to compare the variables and look for a correlation that can help build a better model. Here, we will compare our variables using plots/charts to help us identify the important/useful variables needed for classification.

4.3.1 Species vs bill_length_mm:

```
> ggplot(df, aes(x = bill_length_mm, fill = species))+
+ geom_histogram(position = 'identity', alpha = 0.5)
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
>
```

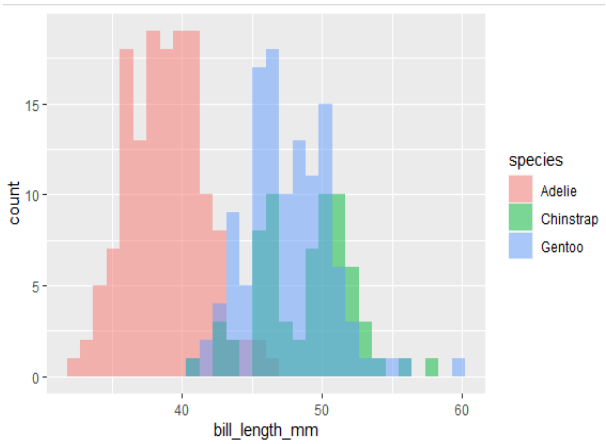

Fig 10: *Histogram of Species and bill length comparison*

Summary: From the histogram above, we can differentiate the Adelie species from the other species through the bill length of the penguin. But it is impossible to differentiate the Chinstrap species from the Gentoo species seeing that they both have longer bills. From the histogram, we can draw the conclusion that Adelie species is known to exhibit shorter bill than the Chinstrap and Gentoo Species.

Let us see if we can differentiate the Gentoo species from the Chinstrap species through the other physical qualities of the penguin.

```
> ggplot(df, aes(x = flipper_length_mm, fill = species))+
+ geom_histogram(position = 'identity', alpha = 0.5)
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
>
```

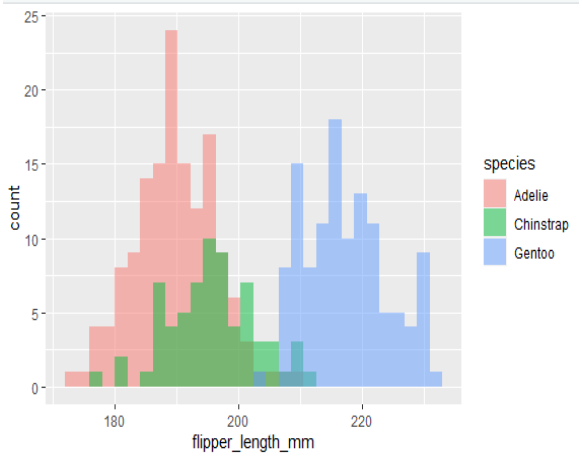

Fig 11*: Histogram of Species and flipper length comparison*

Summary: Flipper length is useful in differentiating the Gentoo species from the Chinstrap species, also useful in differentiating the Adelie species from the Gentoo species but not useful in differentiating the Adelie from the Gentoo species because they both exhibit shorter bills.

4.3.2 Species vs bill depth:

```
> ggplot(df, aes(x = bill_depth_mm, fill = species))+
+ geom_histogram(position = 'identity', alpha = 0.5)
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

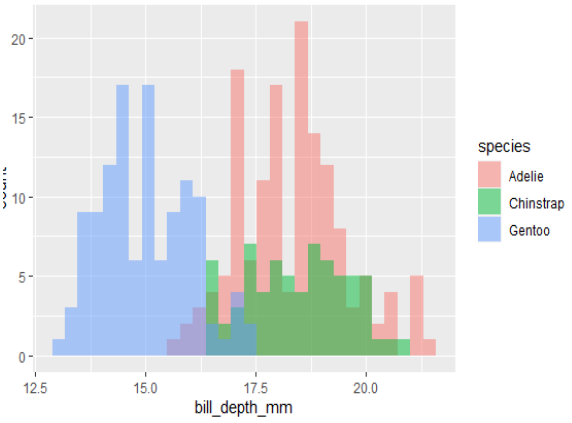

Fig 12: *Histogram of Species and bill depth comparison*

Summary: From the histogram, bill depth is useful in differentiating the Gentoo species from the Chinstrap species, the Adelie species from the Chinstrap species but not useful in differentiating the Adelie species from the Chinstrap species. Adelie species and Chinstrap species both exhibit deeper bills while Gentoo species have shallow bills.

4.3.3 Species vs Weight:

```
> ggplot(df, aes(x = body_mass_g, fill = species))+
+ geom_histogram(position = 'identity', alpha = 0.5)
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
>
```

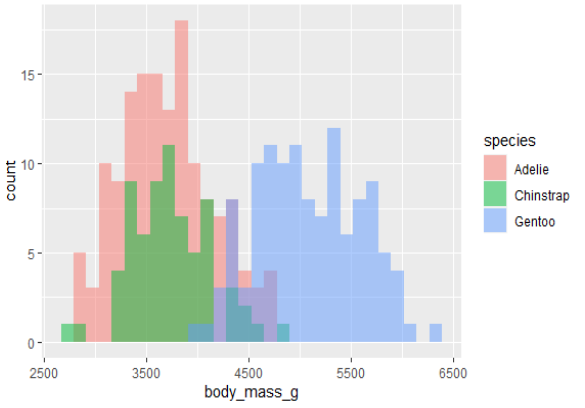

Fig 13*: Histogram of Species and body mass comparison*

Summary: From the histogram, body mass is useful in differentiating the Gentoo species from the Adelie species, the Gentoo species from the Chinstrap species but not useful in differentiating the Adelie species from the Chinstrap species. Adelie species and Chinstrap species both weighs less than the Gentoo species.

4.4 Scatterplot of the Variables:

The Scatterplot just gives us a better understanding/visualization of our dataset. We get to see a better visualization of the variables than with the histogram.

```
> ggplot(df, aes(x = bill_length_mm, y = bill_depth_mm, color = species))+
+ geom_point()
>
```

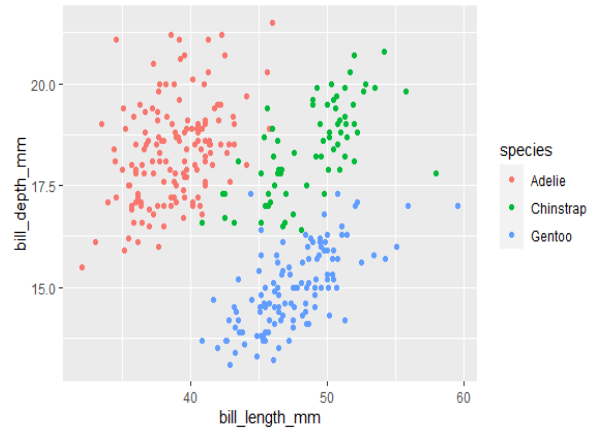

Fig 14: *Scatterplot of bill length and bill depth comparison*

```
> ggplot(df, aes(x = flipper_length_mm, y = body_mass_g, color = species))+
+ geom_point()
>
```

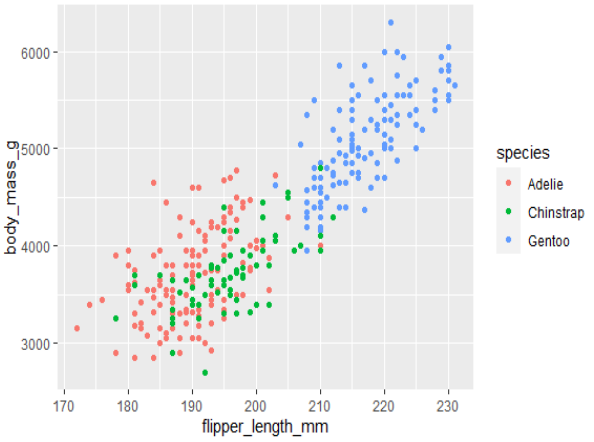

Fig 15: *Scatterplot of flipper length and body mass comparison*

```
> ggplot(df, aes(x = bill_depth_mm, y = body_mass_g, color = species))+
+ geom_point()
>
```

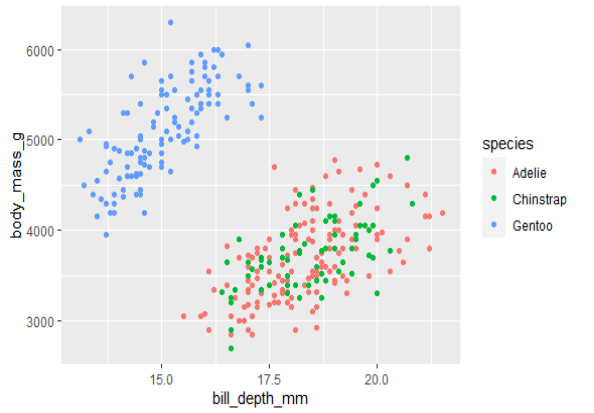

Fig 16: *Scatterplot of bill depth and body mass comparison*

4.4.1 Description of the Scatterplot:

Adelie Species: Deeper bill depth, shorter bill length, short flippers, light weight.

Chinstrap Species: Long bill length, deeper bill depth, short flippers, light weight.

Gentoo Species: Short bill depth, longer bill length, long flippers, heavy weight.

SPECIES	BILL DEPTH	BILL LENGTH	FLIPPER LENGTH	WEIGHT
Adelie	Deep	Short	Short	Light weight
Chinstrap	Deep	Long	Short	Light weight
Gentoo	Shallow	Long	Long	Heavy weight

Table 1: *Table showing description of scatterplot.*

```
> ggplot(df, aes(x = flipper_length_mm, y = bill_length_mm, color = sex))+
+ geom_point()
>
```

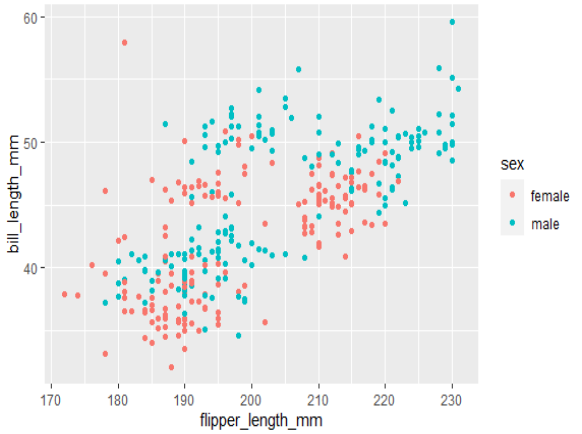

Fig 17: *Scatterplot of bill length and flipper length comparison*

From the scatterplot above, we can see that, we can see that the male has a slightly longer bill and flipper, but this is not conclusive enough and cannot be used to predict if a penguin is male or female.

```
> ggplot(df, aes(x = bill_depth_mm, y = body_mass_g, color = sex))+
+ geom_point()
>
```

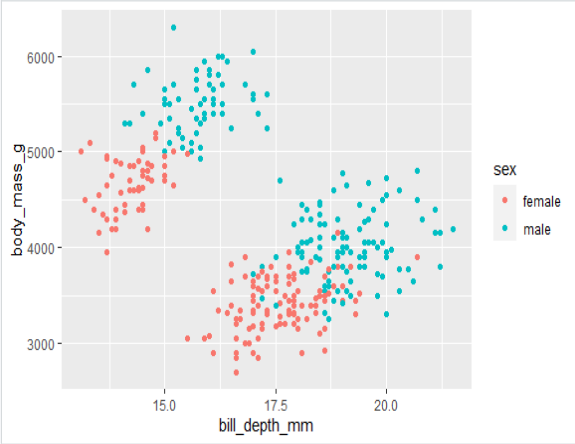

***Fig 18**: Scatterplot of bill depth and body mass comparison based on sex of penguin.*

This chart is still inconclusive in determining the sex of a penguin, the left side of the chart shows some male penguins weigh more than the female penguin, while on the right side of the cluster, the weight of the male penguin Is lesser when compared with the left side. Same goes for the bill length.

Let's get a better visualization by using a bar chart.

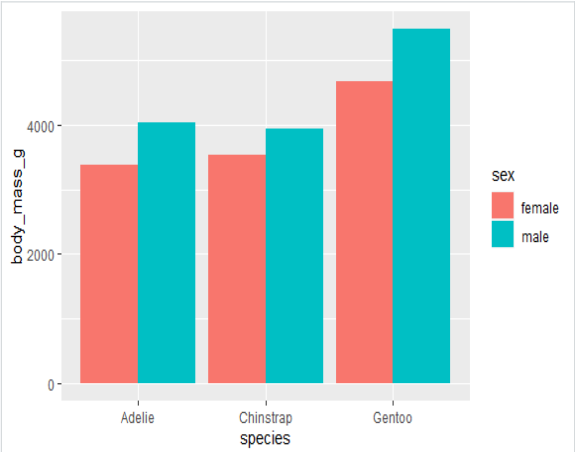

***Fig 19**: Bar chart of species and body mass based on sex of penguin*

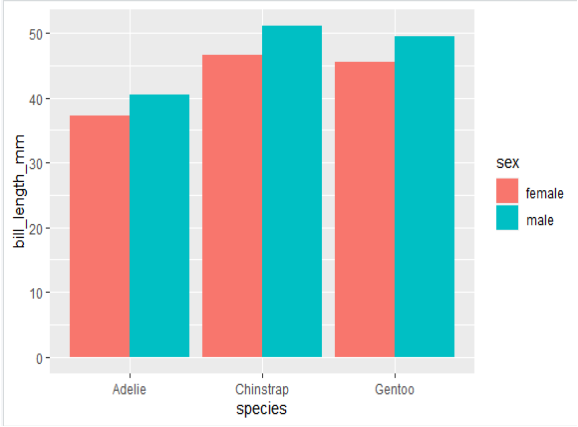

***Fig 20**: Bar chart of species and bill length based on sex of penguin*

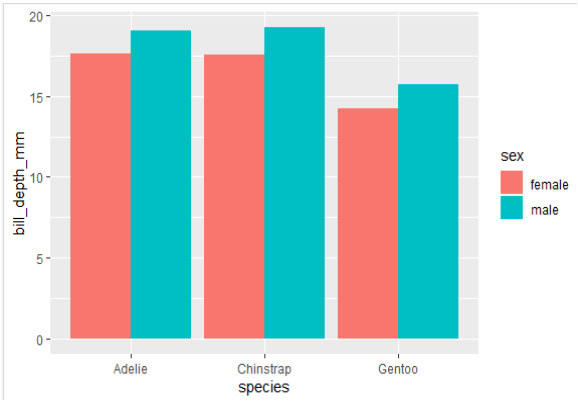

***Fig 21**: Bar chart of species and bill depth based on sex of penguin*

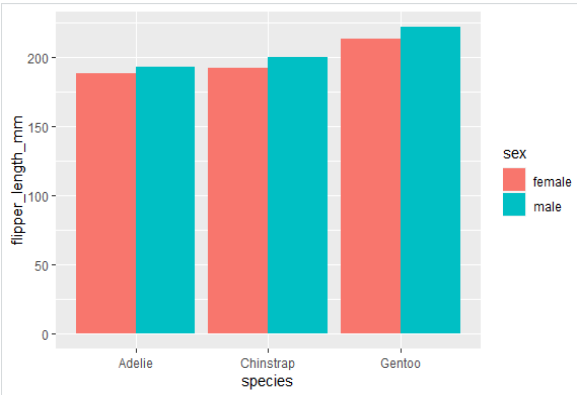

***Fig 22**: Bar chart of species and flipper length based on sex of penguin*

4.5: Correlation Analysis:

A correlation coefficient measures the strength of that relationship.

```
> plot_correlation(na.omit(df), maxcat = 5L)
>
```

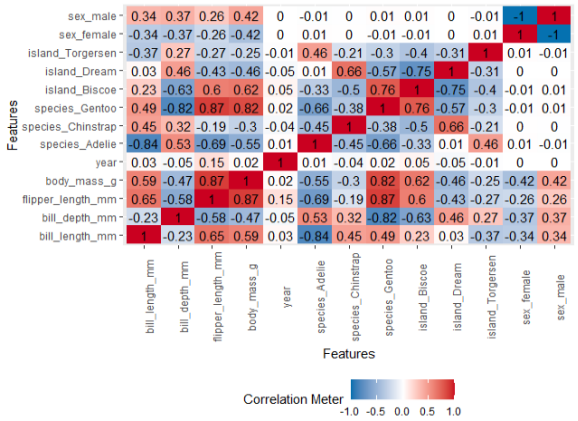

***Fig 23**: Correlation heat map*

Summary of Correlation map:

1. There is a strong correlation between the species Gentoo and the flipper length.
2. There also exist a strong correlation between the species Gentoo and the body mass of the penguin.
3. There is a strong correlation between body mass and flipper length.
4. There is a strong correlation between bill length and flipper length.
5. There is a strong correlation between bill length and body mass.

5 CLASSIFICATION MODELS:

5.1 Classification models:

A classification model tries to draw some conclusion from the input values given for training. It will predict the class labels/categories for the new data. Classification algorithms in machine learning use input training data to predict the likelihood that subsequent data will fall into one of the predetermined categories. In short, classification is a form of "pattern recognition," with classification algorithms applied to the training data to find the same pattern (similar words or sentiments, number sequences, etc.) in future sets of data.

For out dataset, building a classification model to predict the specie of a penguin is straight forward, it is almost perfect seeing that we can tell the species apart through their physical qualities. One thing that is hard to draw a conclusion on is the gender of the penguin. Take for instance, two chinstrap penguins are set apart, similar physical qualities, how can you tell if one is female, and the other is male? It is almost impossible. Under this section, we are going to build a classifier to predict the gender of a penguin based on its physical characteristics.

For this classification model, I will be making use of the caret package in R. I will also be trying out 3 classification methods/algorithm and the comparing the accuracy of each method.

NOTE:

1. Metric considered for this classification is "Accuracy".
2. I will be making use of the linear discriminant analysis, k nearest neighbors, random forest and the SVM method.
3. Experimental setup is the cross-validation method
4. Dataset is partitioned into training and testing groups; A confusion matrix will be used to validate the metric of each method and experimental setup.

Step 1: Remove the insignificant variables, for instance I won't be making use of the year, island and specie columns because there are not important for my prediction. I only need the physical attributes of the penguins for prediction purpose.

```
> View(df)
> main_df <- subset(df, select = -c(year, island, species))
> View(main_df)
>
```

Our main data frame becomes:

	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
1	39.1	18.7	181	3750	male
2	39.5	17.4	186	3800	female
3	40.3	18	195	3250	female
4	36.7	19.3	193	3450	female
5	39.3	20.6	190	3650	male
6	38.9	17.8	181	3625	female
7	39.2	19.6	195	4675	male
8	41.1	17.6	182	3200	female
9	38.6	21.2	191	3800	male
10	34.6	21.1	198	4400	male

ng 1 to 11 of 333 entries, 5 total columns

***Fig 24**: Data frame after removing unimportant columns*

5.2 Creating the partitions (Training data and testing data):

We are going to divide our data set into 2 partitions, 80% for training and the other 20% for validation/testing.

```
> library(caret)
Loading required package: ggplot2
Loading required package: lattice
Warning message:
package 'caret' was built under R version 4.1.3
> training <-createDataPartition(df$sex, p = 0.8, list = FALSE)
> train_data <- df[training,]
> test_data <- df[-training,]
> trc <- trainControl(method = "cv", number = 10)
> metric <- "Accuracy"
```

fit.lda	List of 24
test_data	66 obs. of 5 variables
train_data	267 obs. of 5 variables
training	int [1:267, 1] 3 4 5 6 7 8 9 10 12 14 ...
trc	List of 27
Values	
metric	"Accuracy"

***Fig 25**: Data frame after removing unimportant columns*

The training partition has about 267 observations, while our test data has 66 observations. Next, we are going to train our model using various algorithms and we compare the results of each algorithm using the accuracy metric (This metric is the ratio or the percentage of how many times our model got the predictions correctly). I will train my model using three algorithms (linear regression algorithm, the random forest algorithm, k nearest neighbor (knn) algorithm and the SVM algorithm) in this section and then make predictions using the model, after the predictions have been made, I will the validate my model's prediction using a confusion matrix.

5.3 Linear Discriminant Analysis:

The first model I will be training will make use of the linear discriminant analysis (LDA). LDA is a dimensionality reduction technique that is commonly used for supervised classification problems. It is used for modelling differences in groups i.e., separating two or more classes. It is used to project the features in higher dimension space into a lower dimension space.

STEP 1: Using the train function in the caret package to train our model, I will set the method argument equal to LDA (method = "lda"), this train function will create a model that will predict the gender of the penguins based on the other variables/columns.

```
> set.seed(7)
> fit.lda <- train(sex~., data = df, method="lda", metric=metric, trControl=trc)
```

STEP 2: After my model has been created, I will use the predict function to make predictions based on our newly created model (fit.lda) and our validation data partition which is the test data.

```
> fit.lda <- train(sex~., data = df, method="lda", metric=metric, trControl=trc)
> predictions <- predict(fit.lda, test_data)
> predictions
 [1] male   female female male   female female female female male   male   female female
[13] female male   female male   female male   female female male   female female female
[25] female female female male   male   female male   male   female female female female
[37] male   male   female male   male   female male   male   male   male   female female
[49] male   male   female male   male   male   male   male   male   male   female male
[61] male   male   male   male   male   female
Levels: female male
```

***Fig 26:** Predictions made by our LDA model*

STEP 3: After our model has made its predictions, we are going to test how well our model performed, that is we are going to test the accuracy of each prediction our model made. To do so, I will make use of the confusion Matrix function in the caret package. This confusion matrix gives a summary of the number of correct and incorrect predictions made by the model. It does this by comparing the predictions with the actual result.

```
> confusionMatrix(predictions, test_data$sex)
Confusion Matrix and Statistics

          Reference
Prediction female male
    female     30    1
    male        3   32

               Accuracy : 0.9394
                 95% CI : (0.852, 0.9832)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : 1.042e-14

                  Kappa : 0.8788

 Mcnemar's Test P-Value : 0.6171

            Sensitivity : 0.9091
            Specificity : 0.9697
         Pos Pred Value : 0.9677
         Neg Pred Value : 0.9143
             Prevalence : 0.5000
         Detection Rate : 0.4545
   Detection Prevalence : 0.4697
      Balanced Accuracy : 0.9394

       'Positive' Class : female
```

Fig 27: *Validation result for the lda model*

Summary of model's result: For the female gender, the model made 30 correct predictions and 1 incorrect prediction, while for the male gender, the model made 32 correct predictions and 3 incorrect predictions. Accuracy of the model is about 93.9%, Kappa is about 87.9%. Other parameters such as sensitivity, specificity, detection rate, prevalence etc. are also shown in the result but I will be focusing more on the accuracy of my model when I am comparing it with other classification methods.

NOTE: The caret package basically makes it easy to try out different classification methods with ease and the compare the results of each.

5.4 Random Forest

Random forests or random decision forests is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. The steps involved here are basically the same as above. Build a model, train it, make predictions and then validate our predictions. For the train function, we are going to set our method as random forest (method= "rf").

STEP 1: Training the model.

```
> fit.rf <- train(sex~., data = df, method="rf", metric=metric, trControl=trc)
>
```

STEP 2: Making predictions

```
> set.seed(7)
> fit.rf <- train(sex~., data = df, method="rf", metric=metric, trControl=trc)
> predictions <- predict(fit.rf, test_data$sex)
Error in eval(predvars, data, env) : object 'bill_length_mm' not found
> predictions <- predict(fit.rf, test_data)
> predictions
 [1] male   female female male   female female female female male   male   female female
[13] female male   female male   female male   female female male   female female female
[25] female female female male   male   female male   male   female male   female female
[37] male   male   female male   female female male   male   male   male   female female
[49] male   male   female male   male   female male   male   male   male   female male
[61] male   male   male   female male   female
Levels: female male
>
```

Fig 28: *Predictions made by our RF model*

STEP 3: Results

```
> confusionMatrix(predictions, test_data$sex)
Confusion Matrix and Statistics

          Reference
Prediction female male
    female     33    0
    male        0   33

               Accuracy : 1
                 95% CI : (0.9456, 1)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1

 Mcnemar's Test P-Value : NA

            Sensitivity : 1.0
            Specificity : 1.0
         Pos Pred Value : 1.0
         Neg Pred Value : 1.0
             Prevalence : 0.5
         Detection Rate : 0.5
   Detection Prevalence : 0.5
      Balanced Accuracy : 1.0

       'Positive' Class : female
```

Fig 29: *Validation result for the rf model*

Summary of model's result: For the female gender, the model made 33 correct predictions and no incorrect prediction, while for the male gender, the model made 33 correct predictions and 0 incorrect predictions. Accuracy of the model is 100%, Kappa is about 100%. Other parameters such as sensitivity, specificity (both 100% as well), detection rate, prevalence etc. are also shown in the result.

5.5 K Nearest Neighbor (KNN):

The k-nearest neighbors (KNN) algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems. It's easy to implement and understand but has a major drawback of becoming significantly slows as the size of that data in use grows. The steps involved here are basically the same as above. Build a model, train it, make predictions and then validate our predictions. For the train function, we are going to set our method as random forest (method=" knn").

STEP 1: Training the model.

```
> set.seed(7)
> fit.knn <- train(sex~., data = df, method="knn", metric=metric, trControl=trc)
```

STEP 2: Making predictions.

```
> set.seed(7)
> fit.knn <- train(sex~., data = df, method="knn", metric=metric, trControl=trc)
> predictions <- predict(fit.knn, test_data)
> predictions
 [1] male   female female male   female female female female male   male   female female
[13] female male   female male   female male   female female male   female female female
[25] female female female male   male   female male   male   female female female female
[37] male   male   female female male   female male   male   male   male   female female
[49] male   female female male   male   female male   female female female female male
[61] male   female male   male   male   female
Levels: female male
```

Fig 30: *Predictions made by our KNN model*

STEP 3: Results

```
> confusionMatrix(predictions, test_data$sex)
Confusion Matrix and Statistics

          Reference
Prediction female male
    female     31    7
    male        2   26

               Accuracy : 0.8636
                 95% CI : (0.7569, 0.9357)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : 5.914e-10

                  Kappa : 0.7273

 Mcnemar's Test P-Value : 0.1824

            Sensitivity : 0.9394
            Specificity : 0.7879
         Pos Pred Value : 0.8158
         Neg Pred Value : 0.9286
             Prevalence : 0.5000
         Detection Rate : 0.4697
   Detection Prevalence : 0.5758
      Balanced Accuracy : 0.8636

       'Positive' Class : female
```

Fig 31: *Validation result for the KNN model*

Summary of model's result: For the female gender, the model made 31 correct predictions and 7 incorrect predictions, while for the male gender, the model made 26 correct predictions and 2 incorrect predictions. Accuracy of the model is 86.4%, Kappa is about 72.7%. Other parameters such as sensitivity (93.9%), specificity (78.7%), detection rate (46.9%), prevalence (50%) etc. are also shown in the result.

5.6 Support Vector Machines (SVM):

A support vector machine (SVM) uses algorithms to train and classify data within degrees of polarity, taking it to a degree beyond *X/Y* prediction.

STEP 1: Model fitting and training

```
> set.seed(7)
> fit.svm <- train(sex~., data = df, method="svmRadial", metric=metric, trControl=trc)
>
```

STEP 2: Making predictions

```
> set.seed(7)
> fit.svm <- train(sex~., data = df, method="svmRadial", metric=metric, trControl=trc)
> predictions <- predict(fit.SVM, test_data)
Error in predict(fit.SVM, test_data) : object 'fit.SVM' not found
> predictions <- predict(fit.svm, test_data)
> predictions
 [1] male   female female male   female female female female male   male   female female
[13] female male   female male   female male   female female male   female female female
[25] female female female male   male   female male   male   female female female female
[37] male   male   female male   male   female male   male   male   female male
[49] male   male   female male   male   female male   male   male   female male
[61] male   male   male   male   male   female
Levels: female male
>
```

Fig 32: *Predictions made by our SVM model*

STEP 3: Results

```
> confusionMatrix(predictions, test_data$sex)
Confusion Matrix and Statistics

          Reference
Prediction female male
    female     30    1
    male        3   32

               Accuracy : 0.9394
                 95% CI : (0.852, 0.9832)
    No Information Rate : 0.5
    P-Value [Acc > NIR] : 1.042e-14

                  Kappa : 0.8788

 Mcnemar's Test P-Value : 0.6171

            Sensitivity : 0.9091
            Specificity : 0.9697
         Pos Pred Value : 0.9677
         Neg Pred Value : 0.9143
             Prevalence : 0.5000
         Detection Rate : 0.4545
   Detection Prevalence : 0.4697
      Balanced Accuracy : 0.9394

       'Positive' Class : female

>
```

Fig 33: *Validation result for the SVM model*

Summary of model's result: For the female gender, the model made 30 correct predictions and 1 incorrect prediction, while for the male gender, the model made 32 correct predictions and 3 incorrect predictions. Accuracy of the model is 93.9%, Kappa is about 87.9%. Other parameters such as sensitivity (90.9%), specificity (96.9%), detection rate (45.5%), prevalence (50%) etc. are also shown in the result.

Now let us compare the results of each model.

6 RESULTS

6.1 Comparison of our models:

Our models did well in predicting the gender of the penguins, but one model was the best and did way better than the others in terms of the metrics I used. Under this section, I will show a graphical comparison of all the model used for our analysis.

Summary of each result is shown below.

```
> comparison_result <- resamples(list(lda=fit.lda, rf=fit.rf, knn=fit.knn, svm=fit.svm))
> summary(comparison_result)

Call:
summary.resamples(object = comparison_result)

Models: lda, rf, knn, svm
Number of resamples: 10

Accuracy
        Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
lda 0.8437500 0.8584559 0.8957219 0.8946914 0.9110963 0.9705882    0
rf  0.8529412 0.8890374 0.9090909 0.9159258 0.9411765 0.9705882    0
knn 0.6562500 0.7132353 0.7845644 0.7799354 0.8221925 0.9411765    0
svm 0.8235294 0.8823529 0.9090909 0.9008523 0.9303977 0.9705882    0

Kappa
        Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
lda 0.6875000 0.7169118 0.7908557 0.7892707 0.8222342 0.9411765    0
rf  0.7058824 0.7777808 0.8183486 0.8317673 0.8823529 0.9411765    0
knn 0.3125000 0.4264706 0.5700983 0.5600174 0.6449657 0.8823529    0
svm 0.6470588 0.7647059 0.8173426 0.8015534 0.8608372 0.9411765    0

>
```

Fig 34: *Summary of each model's result*

A dot plot of the result is given below.

```
> dotplot(comparison_result)
>
```

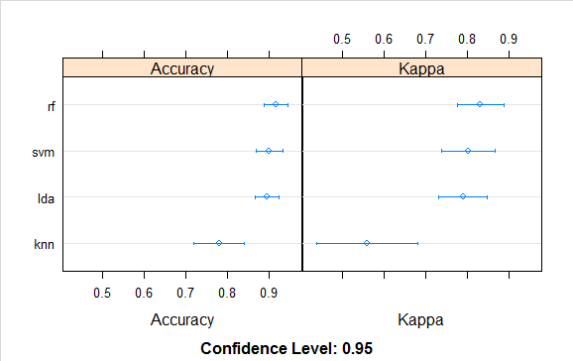

Fig 35: *Dot plot showing comparison of each model*

Summary of dot plot: Based on accuracy and kappa, the random forest algorithm performed better than the rest algorithms, followed by the support vector machines, then followed by the linear discriminant analysis and the k nearest neighbor performed the least in both metrics.

6.2 Variable importance:

Variable importance evaluation functions can be separated into two groups: those that use the model information and those that do not. The advantage of using a model-based approach is that is more closely tied to the model performance and that it *may* be able to incorporate the correlation structure between the predictors into the importance calculation.

Using the caret package, we can also show the variables that played a very important role in each model. The "varimp function" can help us check which variables were significantly important in each model.

For this section, we will focus on the best model and check the most important variable in the prediction.

Random Forest:

```
> importance <- varImp(fit.rf)
> plot(importance)
>
```

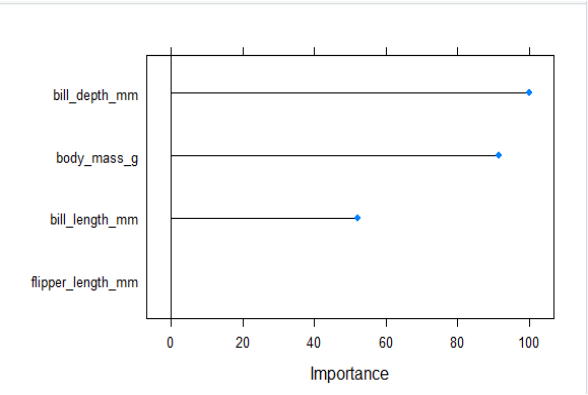

Fig 34: *Summary of each model's result*

Summary of plot: We can see that flipper length played no role in prediction in the random forest model. Bill depth (100) and body mass (about 97) played a very significant role in predicting the gender of a penguin using the random forest classifier. Bill length played an average role as well.

7 CONCLUSIONS

From our analysis, several conclusions can be drawn

1. Our of the four classifiers experimented on, based on the accuracy and kappa metric, the random forest model performed the best over the others in predicting the gender of the penguin, while the KNN model was the least accurate.
2. From the random forest model, the major determinant in determining the gender of a penguin was the bill depth and body mass.
3. The flipper length was insignificant in determining the gender of the penguin.
4. Random forest and SVM models are more efficient in making predictions.
5. We can determine the species of a penguin through its physical characteristics.