Project Description

This project is a classification task, whose purpose is to classify the iris flowers into their respective species based on available attributes, or in other words, predict the flower class based on available attributes. The three iris species include Setosa, Versicolor and Virginica, and the explanatory variables (attributes) include sepal length, sepal width, pedal length and petal width. The next picture shows an example of the petal and sepal lengths and widths.

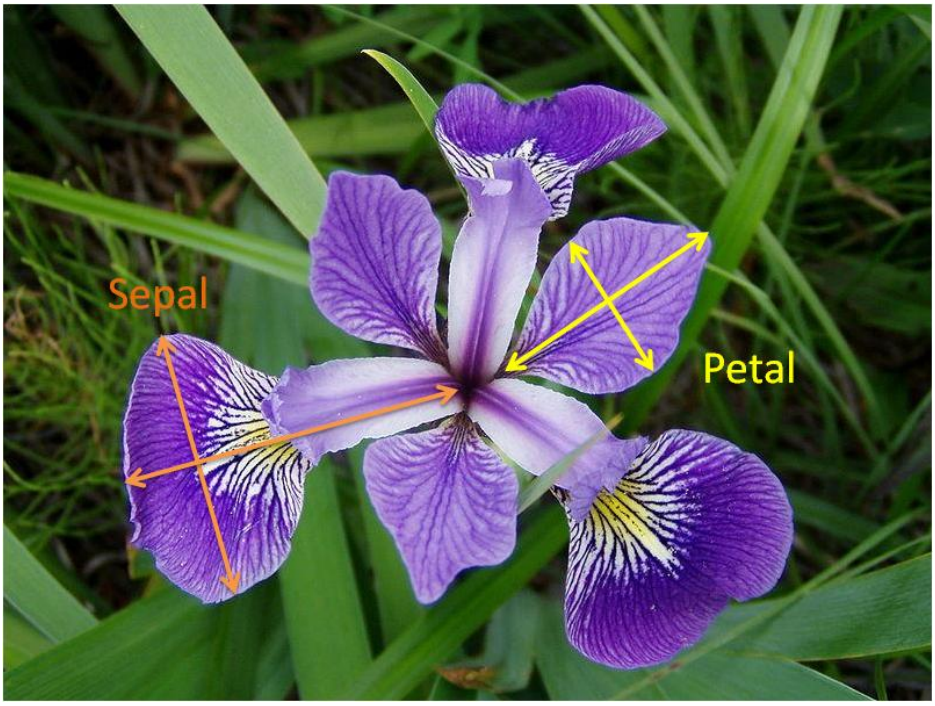

Figure 1: Petal and sepal lengths and widths.

Descriptive Statistics and Exploratory Analysis

First, we are going to take a look at the data. The dataset contains 150 observations of iris flowers. There are four columns of measurements of the flowers in centimeters. The fifth column is the species of the flower observed. All observed flowers belong to one of the three species (Setosa, Versicolor or Virginica). Specifically, we can see that each class has the same number of instances (50 or 33% of the dataset).

	freq	percentage
Iris-setosa	50	33.33333
Iris-versicolor	50	33.33333
Iris-virginica	50	33.33333

The next table shows a summary of each attribute, which includes the mean, the median, the min and max values and the 1st and 3rd quartiles.

```
 Sepal.Length    Sepal.Width     Petal.Length    Petal.Width      Species
Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   Length:150
1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   Class :character
Median :5.800   Median :3.000   Median :4.350   Median :1.300   Mode  :character
Mean   :5.843   Mean   :3.054   Mean   :3.759   Mean   :1.199
3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```

Now, we are going to use some visualizations in order to better understand the relationships between the attributes. For this purpose, we create two scatter plots: one of the petal length and the petal width, and one of the sepal length and the sepal width. As we can see from the figures, Setosa seems to be the most distinguishable of the three species with respect to both petal and sepal attributes.

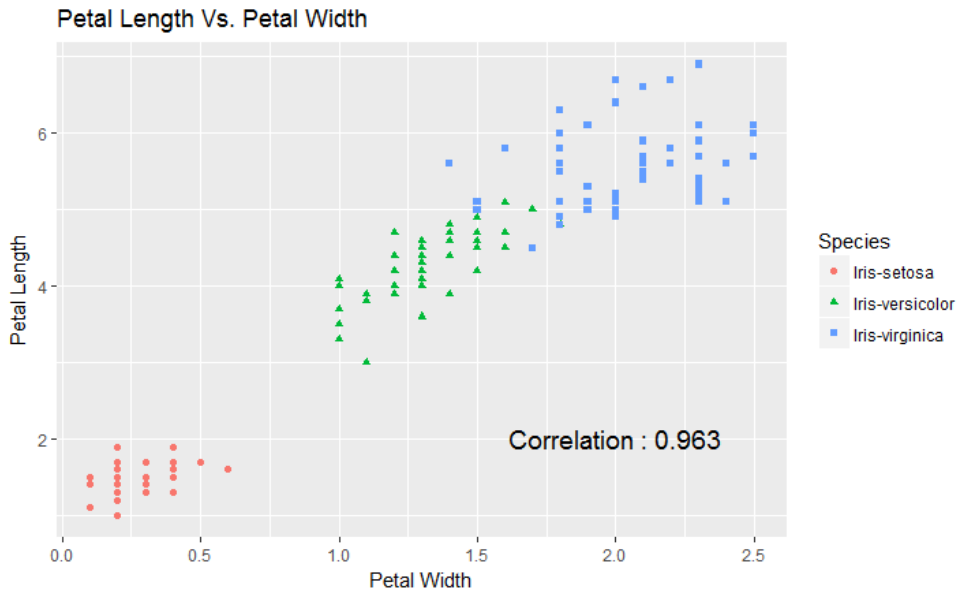

Figure 2: Scatter plot of petal length versus petal width.

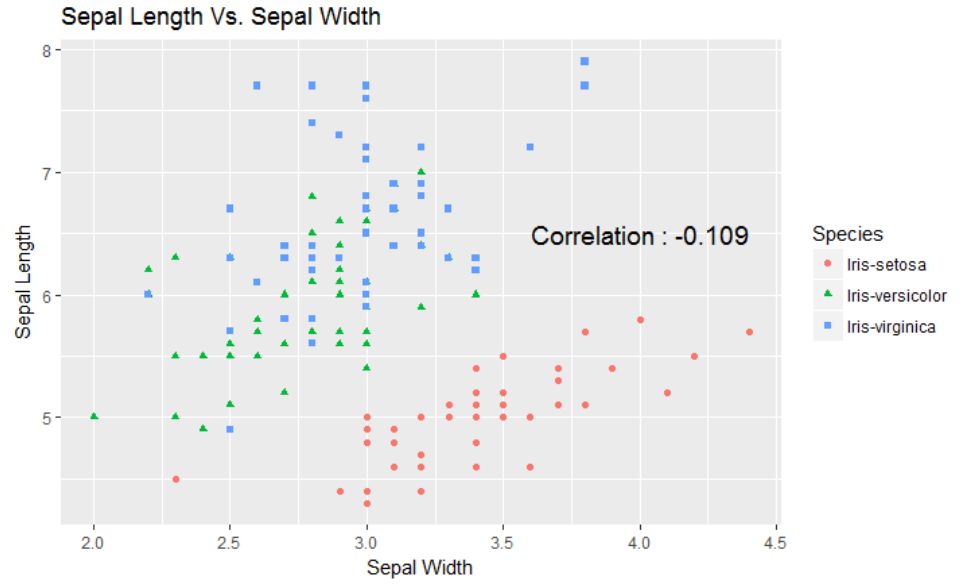

Figure 3: Scatter plot of sepal length versus sepal width.

The next figure shows the relationships across all available attributes.

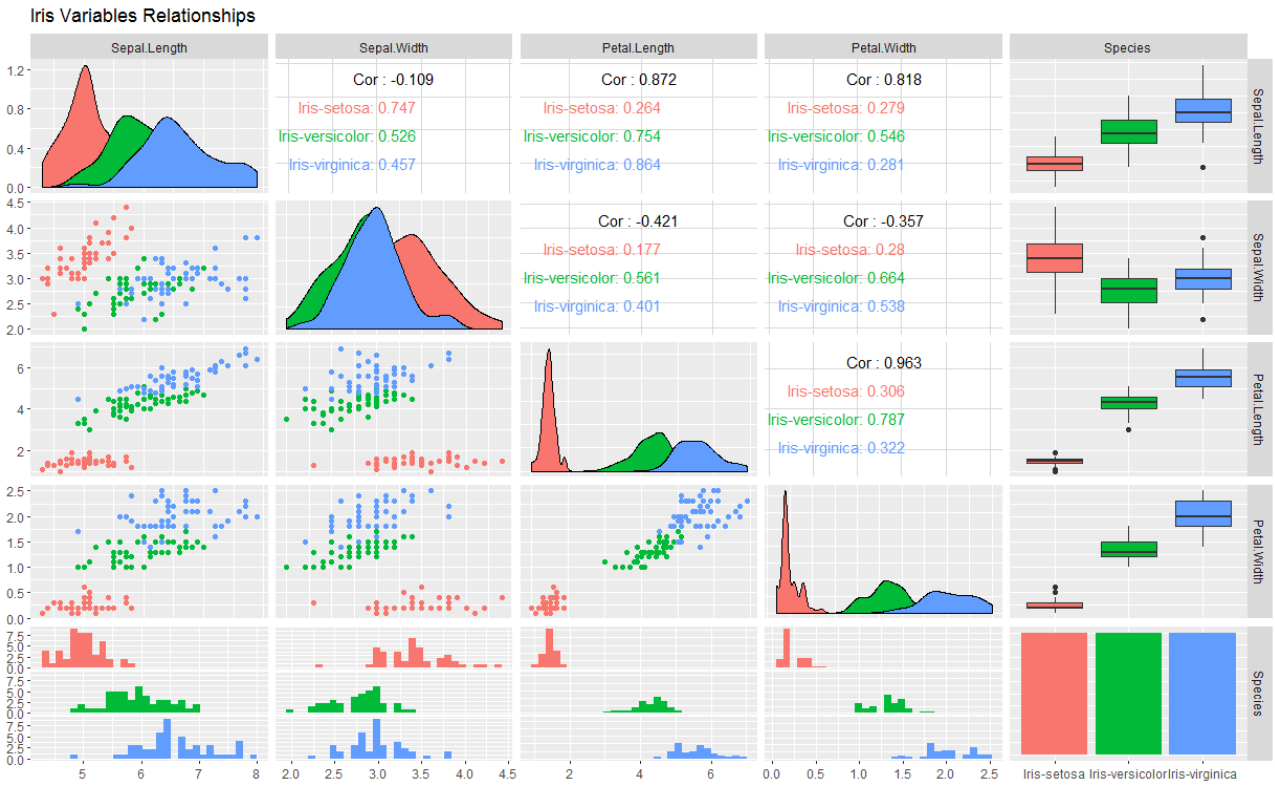

Figure 4: Matrix of plots that shows the relationships across all attributes.

Splitting the Data into Training and Testing Sets

Before creating some models of the data, we need to split the dataset into training set and testing set. In this project, we used 75% of the data (chosen randomly) to build and train the models and the remaining 25% to assess how well the developed algorithms work. That is, we are going to use the training set in order to understand the data, select the appropriate model and determine the model parameters, and the testing set, which contains unseen data, in order to get a realistic and more concrete estimate of the models' performance.

Model Building

Now, we will build and fit some models to the training set and we will estimate their accuracy on the testing set. For this purpose, we will build eight different models: K-Means Clustering, Linear Discriminant Analysis (LDA), Decision Tree (CART), Random Forest (RF), Gradient Boosting Method (GBM), k-Nearest Neighbor (kNN), Support Vector Machines (SVM) with Radial Basis Function Kernel and Neural Network (NN). Two of them are linear (LDA, K-Means Clustering), two are nonlinear (CART, kNN) and four are complex nonlinear (RF, GBM, SVM, NN). We reset the random number seed before each run to ensure that each algorithm will be evaluated using the same data partitions. This means that the results will be directly comparable.

k-Means Clustering

k-means algorithm is one of the simplest unsupervised learning algorithms. Its goal is to optimally divide a set of data points into k groups (or clusters), so that the total distance between the cluster's members and its representative (or centroid) is minimized, while the distance between different clusters is maximized [1].

Since we know there are 3 classes, we begin with 3 centers. Also, since k-means assigns the centroids randomly, we specify nstart as 20 to run the algorithm 20 times with 20 random starting sets of centroids and then pick the best of those 20.

Now we can check how k-means algorithm performs on the testing set. Below we can see the results from the terminal output.

```
  Iris-setosa Iris-versicolor Iris-virginica
1          50               0              0
2           0               2             36
3           0              48             14
```

We see that by using k-means algorithm, we misclassify sixteen observations (two from the Versicolor specie and fourteen from Virginica).

We can also plot the clusters and their centroids to see how the algorithm clustered the observations.

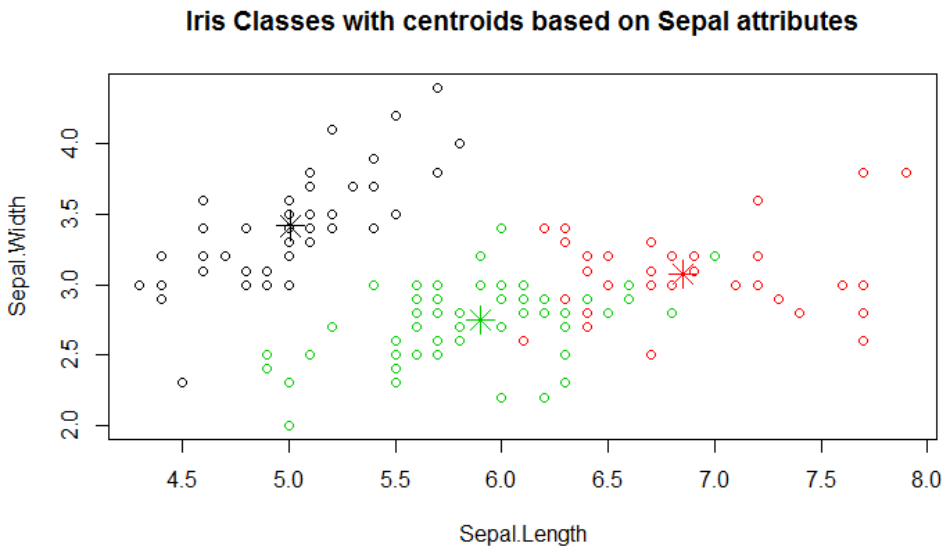

Figure 5: Iris classes with centroids based on sepal attributes, as they were computed by k-means algorithm.

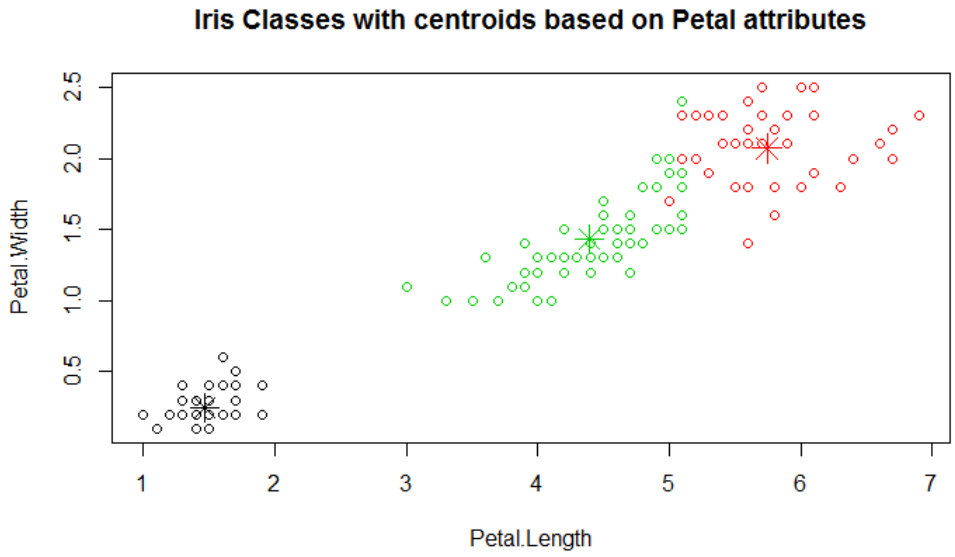

Figure 6: Iris classes with centroids based on petal attributes, as they were computed by k-means algorithm.

Linear Discriminant Analysis (LDA)

LDA classifier basically projects the data onto a lower-dimensional vector space such that the ratio of the between-class-distance to the within-class-distance is maximized, thus achieving maximum discrimination [2]. Obviously the classifier will reach high accuracy if the data are linear separable.

First, we can check how LDA performs on the training set. Below we can see the results from the terminal output.

```
Confusion Matrix and Statistics

                 Reference
Prediction        Iris-setosa Iris-versicolor Iris-virginica
  Iris-setosa              38               0              0
  Iris-versicolor           0              37              1
  Iris-virginica            0               1             37

Overall Statistics

               Accuracy : 0.9825
                 95% CI : (0.9381, 0.9979)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9737
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Iris-setosa Class: Iris-versicolor Class: Iris-virginica
Sensitivity                      1.0000                 0.9737                0.9737
Specificity                      1.0000                 0.9868                0.9868
Pos Pred Value                   1.0000                 0.9737                0.9737
Neg Pred Value                   1.0000                 0.9868                0.9868
Prevalence                       0.3333                 0.3333                0.3333
Detection Rate                   0.3333                 0.3246                0.3246
Detection Prevalence             0.3333                 0.3333                0.3333
Balanced Accuracy                1.0000                 0.9803                0.9803
```

Now, we can check how LDA performs on the testing set. Below we can see the results from the terminal output.

```
Confusion Matrix and Statistics

                 Reference
Prediction        Iris-setosa Iris-versicolor Iris-virginica
  Iris-setosa              12               0              0
  Iris-versicolor           0              11              0
  Iris-virginica            0               1             12

Overall Statistics

               Accuracy : 0.9722
                 95% CI : (0.8547, 0.9993)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 4.864e-16

                  Kappa : 0.9583
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Iris-setosa Class: Iris-versicolor Class: Iris-virginica
Sensitivity                      1.0000                 0.9167                 1.0000
Specificity                      1.0000                 1.0000                 0.9583
Pos Pred Value                   1.0000                 1.0000                 0.9231
Neg Pred Value                   1.0000                 0.9600                 1.0000
Prevalence                       0.3333                 0.3333                 0.3333
Detection Rate                   0.3333                 0.3056                 0.3333
Detection Prevalence             0.3333                 0.3056                 0.3611
Balanced Accuracy                1.0000                 0.9583                 0.9792
```

We see that by using this algorithm, we misclassify only one observation (from the Versicolor specie). Also, we observe that, as it was expected, the accuracy on the test set is worse than the accuracy on the training set. Note that if the accuracy of the training set differs from that of the testing set by a large value, it is usually an indication of overfitting.

Decision Tree (CART)

Decision Trees classify observations by sorting them down the tree from the root node to some leaf node which provides the classification of the observation. Each node in the tree specifies a test on a particular attribute of the observation, and each branch descending from that node corresponds to one of the possible values for that test [3]. In this project, we will use the rpart (which stands for recursive partitioning) algorithm to build the tree. The complexity parameter (cp) is used to control the size of the decision tree and to select the optimal tree size. In order to find the optimal value for cp, we use bootstrap method.

```
CART

114 samples
  4 predictor
  3 classes: 'Iris-setosa', 'Iris-versicolor', 'Iris-virginica'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 114, 114, 114, 114, 114, 114, ...
Resampling results across tuning parameters:

  cp         Accuracy   Kappa
  0.0000000  0.9516993  0.9269501
  0.4605263  0.7225166  0.5924572
```

```
  0.5000000   0.5116150   0.2996360

Accuracy was used to select the optimal model using  the largest value.
The final value used for the model was cp = 0.
```

The terminal output above shows that, based on accuracy, our optimal cp is 0, which actually means to not prune the tree.

The next figure illustrates the decision tree. As we can see, Iris Setosa can be successfully isolated from other species by petal length. On the other hand, Iris Versicolor and Iris Virgicina are somewhat more difficult to separate based on petal width.

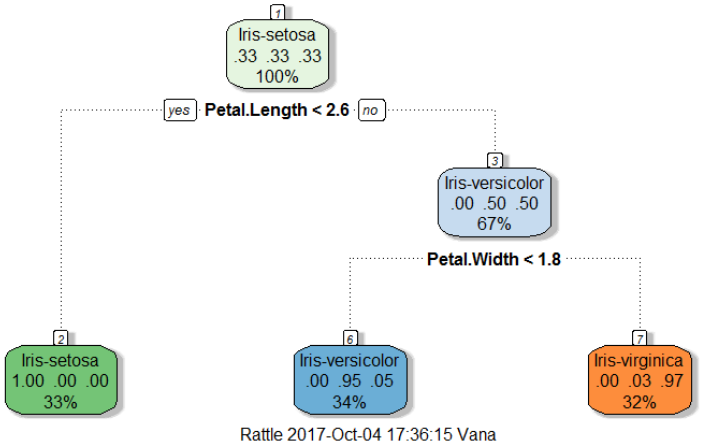

Figure 7: Visualization of the decision tree.

First, we can check how Decision Tree performs on the training set. Below we can see the results from the terminal output.

```
Confusion Matrix and Statistics

                 Reference
Prediction        Iris-setosa Iris-versicolor Iris-virginica
  Iris-setosa              38               0              0
  Iris-versicolor           0              37              2
  Iris-virginica            0               1             36

Overall Statistics

               Accuracy : 0.9737
                 95% CI : (0.925, 0.9945)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9605
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Iris-setosa Class: Iris-versicolor Class: Iris-virginica
Sensitivity                      1.0000                 0.9737                0.9474
Specificity                      1.0000                 0.9737                0.9868
Pos Pred Value                   1.0000                 0.9487                0.9730
Neg Pred Value                   1.0000                 0.9867                0.9740
Prevalence                       0.3333                 0.3333                0.3333
Detection Rate                   0.3333                 0.3246                0.3158
Detection Prevalence             0.3333                 0.3421                0.3246
Balanced Accuracy                1.0000                 0.9737                0.9671
```

Now, we can check how Decision Tree performs on the testing set. Below we can see the results from the terminal output.

```
Confusion Matrix and Statistics

                 Reference
Prediction        Iris-setosa Iris-versicolor Iris-virginica
  Iris-setosa              12               0              0
  Iris-versicolor           0              12              3
  Iris-virginica            0               0              9

Overall Statistics

               Accuracy : 0.9167
                 95% CI : (0.7753, 0.9825)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 3.978e-13

                  Kappa : 0.875
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Iris-setosa Class: Iris-versicolor Class: Iris-virginica
Sensitivity                      1.0000                 1.0000                0.7500
Specificity                      1.0000                 0.8750                1.0000
Pos Pred Value                   1.0000                 0.8000                1.0000
Neg Pred Value                   1.0000                 1.0000                0.8889
Prevalence                       0.3333                 0.3333                0.3333
Detection Rate                   0.3333                 0.3333                0.2500
Detection Prevalence             0.3333                 0.4167                0.2500
Balanced Accuracy                1.0000                 0.9375                0.8750
```

We see that, by using Decision Trees, we misclassify three observations from the Virginica specie.

Random Forest (RF)

Random Forest (RF) is an algorithm that fits many classification (or regression) tree (CART) models to random subsets of the input data and averages the predictions from such simple trees for prediction [4]. Random forests correct for decision trees' habit of overfitting to their training set [5].

The optimal number of variables used at each split of the tree (mtry parameter) is determined with the bootstrap method.

```
Random Forest

114 samples
  4 predictor
  3 classes: 'Iris-setosa', 'Iris-versicolor', 'Iris-virginica'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 114, 114, 114, 114, 114, 114, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
  2     0.9530165  0.9286428
  3     0.9551032  0.9318577
  4     0.9523677  0.9277701

Accuracy was used to select the optimal model using  the largest value.
The final value used for the model was mtry = 3.
```

The terminal output above shows that, based on accuracy, our optimal mtry is 3.

First, we can check how Random Forest performs on the training set. Below we can see the results from the terminal output.

```
Confusion Matrix and Statistics

                 Reference
Prediction        Iris-setosa Iris-versicolor Iris-virginica
  Iris-setosa              38               0              0
  Iris-versicolor           0              38              0
  Iris-virginica            0               0             38

Overall Statistics

               Accuracy : 1
                 95% CI : (0.9682, 1)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Iris-setosa Class: Iris-versicolor Class: Iris-virginica
Sensitivity                      1.0000                 1.0000                1.0000
Specificity                      1.0000                 1.0000                1.0000
Pos Pred Value                   1.0000                 1.0000                1.0000
Neg Pred Value                   1.0000                 1.0000                1.0000
Prevalence                       0.3333                 0.3333                0.3333
Detection Rate                   0.3333                 0.3333                0.3333
Detection Prevalence             0.3333                 0.3333                0.3333
Balanced Accuracy                1.0000                 1.0000                1.0000
```

It's worth noting that the Random Forest algorithm achieves perfect classification accuracy on the training set. Now, we can check how Random Forest performs on the testing set. Below we can see the results from the terminal output.

```
Confusion Matrix and Statistics

                 Reference
Prediction        Iris-setosa Iris-versicolor Iris-virginica
  Iris-setosa              12               0              0
  Iris-versicolor           0              11              1
  Iris-virginica            0               1             11

Overall Statistics

               Accuracy : 0.9444
                 95% CI : (0.8134, 0.9932)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 1.728e-14

                  Kappa : 0.9167
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Iris-setosa Class: Iris-versicolor Class: Iris-virginica
Sensitivity                      1.0000                 0.9167                0.9167
Specificity                      1.0000                 0.9583                0.9583
Pos Pred Value                   1.0000                 0.9167                0.9167
Neg Pred Value                   1.0000                 0.9583                0.9583
Prevalence                       0.3333                 0.3333                0.3333
Detection Rate                   0.3333                 0.3056                0.3056
Detection Prevalence             0.3333                 0.3333                0.3333
Balanced Accuracy                1.0000                 0.9375                0.9375
```

We see that although the accuracy on the training set was perfect and better comparing with that of the Decision Trees algorithm, we still misclassify two observations (one from the Versicolor specie and one from Virginica).

Gradient Boosting Method (GBM)

The way the Gradient Boosting Method (GBM) works is very similar to the Random Forest, since both approaches combine weaker models (typically decision trees) to predict the class. Their difference is that Random Forest trains the trees from different random subsets of the input data, while Gradient Boosting takes the error from the previous tree and uses it to improve the next one [6].

In GBM, we use the bootstrap method to fine tune several parameters such as n.trees (number of trees), interaction.depth (maximum nodes per tree or number of splits it has to perform on a tree), shrinkage (learning rate, which is used for reducing, or shrinking, the impact of each additional tree) and n.minobsinnode (minimum number of observations in trees' terminal nodes).

```
Stochastic Gradient Boosting

114 samples
  4 predictor
  3 classes: 'Iris-setosa', 'Iris-versicolor', 'Iris-virginica'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 114, 114, 114, 114, 114, 114, ...
Resampling results across tuning parameters:

  interaction.depth  n.trees  Accuracy   Kappa
  1                   50      0.9553133  0.9323587
  1                  100      0.9562250  0.9337074
  1                  150      0.9570530  0.9348769
  2                   50      0.9509340  0.9256638
  2                  100      0.9561584  0.9336121
  2                  150      0.9532532  0.9291326
  3                   50      0.9551939  0.9320881
  3                  100      0.9532743  0.9291139
  3                  150      0.9512176  0.9259860

Tuning parameter 'shrinkage' was held constant at a value of 0.1
Tuning parameter 'n.minobsinnode' was
 held constant at a value of 10
Accuracy was used to select the optimal model using  the largest value.
The final values used for the model were n.trees = 150, interaction.depth = 1, shrinkage
= 0.1 and
 n.minobsinnode = 10.
```

The terminal output above shows that, based on accuracy, the optimal values are: n.trees = 150, interaction.depth = 1, shrinkage = 0.1 and n.minobsinnode = 10.

First, we can check how Gradient Boosting Method performs on the training set. Below we can see the results from the terminal output.

```
Confusion Matrix and Statistics

                 Reference
Prediction        Iris-setosa Iris-versicolor Iris-virginica
  Iris-setosa              38               0              0
  Iris-versicolor           0              38              0
  Iris-virginica            0               0             38

Overall Statistics

               Accuracy : 1
                 95% CI : (0.9682, 1)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Iris-setosa Class: Iris-versicolor Class: Iris-virginica
Sensitivity                      1.0000                 1.0000                1.0000
Specificity                      1.0000                 1.0000                1.0000
Pos Pred Value                   1.0000                 1.0000                1.0000
Neg Pred Value                   1.0000                 1.0000                1.0000
Prevalence                       0.3333                 0.3333                0.3333
Detection Rate                   0.3333                 0.3333                0.3333
Detection Prevalence             0.3333                 0.3333                0.3333
Balanced Accuracy                1.0000                 1.0000                1.0000
```

It's worth noting that the Gradient Boosting Method algorithm achieves perfect classification accuracy on the training set. Now, we can check how Gradient Boosting Method performs on the testing set. Below we can see the results from the terminal output.

```
Confusion Matrix and Statistics

                 Reference
Prediction        Iris-setosa Iris-versicolor Iris-virginica
  Iris-setosa              12               0              0
  Iris-versicolor           0              11              1
  Iris-virginica            0               1             11

Overall Statistics

               Accuracy : 0.9444
                 95% CI : (0.8134, 0.9932)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 1.728e-14

                  Kappa : 0.9167
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Iris-setosa Class: Iris-versicolor Class: Iris-virginica
Sensitivity                      1.0000                 0.9167                0.9167
Specificity                      1.0000                 0.9583                0.9583
Pos Pred Value                   1.0000                 0.9167                0.9167
Neg Pred Value                   1.0000                 0.9583                0.9583
Prevalence                       0.3333                 0.3333                0.3333
Detection Rate                   0.3333                 0.3056                0.3056
Detection Prevalence             0.3333                 0.3333                0.3333
Balanced Accuracy                1.0000                 0.9375                0.9375
```

We see that although the accuracy on the training set was perfect and the algorithm is more sophisticated compared to Random Forest, we still misclassify two observations (one from the Versicolor specie and one from Virginica).

k-Nearest Neighbor (kNN)

In the kth-Nearest Neighbor (kNN) algorithm, an observation is classified by a majority vote of its k closest points (neighbors), determined by a predefined distance function. The observation is then assigned to the class most common amongst its k nearest neighbors. The number "k" is the nearest neighbors we wish to take vote from and is typically a small positive integer [7]. In this project, we use bootstrap to find the optimal value for k.

```
k-Nearest Neighbors

114 samples
  4 predictor
  3 classes: 'Iris-setosa', 'Iris-versicolor', 'Iris-virginica'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 114, 114, 114, 114, 114, 114, ...
Resampling results across tuning parameters:

  k  Accuracy   Kappa
  5  0.9687057  0.9525018
  7  0.9695206  0.9538723
  9  0.9713767  0.9567316

Accuracy was used to select the optimal model using  the largest value.
The final value used for the model was k = 9.
```

The terminal output above shows that, based on accuracy, our optimal k is 9.

First, we can check how kNN performs on the training set. Below we can see the results from the terminal output.

```
Confusion Matrix and Statistics

                 Reference
Prediction        Iris-setosa Iris-versicolor Iris-virginica
  Iris-setosa              38               0              0
  Iris-versicolor           0              36              0
  Iris-virginica            0               2             38

Overall Statistics

               Accuracy : 0.9825
                 95% CI : (0.9381, 0.9979)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9737
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Iris-setosa Class: Iris-versicolor Class: Iris-virginica
Sensitivity                      1.0000                 0.9474                1.0000
Specificity                      1.0000                 1.0000                0.9737
Pos Pred Value                   1.0000                 1.0000                0.9500
Neg Pred Value                   1.0000                 0.9744                1.0000
Prevalence                       0.3333                 0.3333                0.3333
Detection Rate                   0.3333                 0.3158                0.3333
Detection Prevalence             0.3333                 0.3158                0.3509
Balanced Accuracy                1.0000                 0.9737                0.9868
```

Now, we can check how kNN performs on the testing set. Below we can see the results from the terminal output.

```
Confusion Matrix and Statistics

                 Reference
Prediction        Iris-setosa Iris-versicolor Iris-virginica
  Iris-setosa              12               0              0
  Iris-versicolor           0              11              1
  Iris-virginica            0               1             11

Overall Statistics

               Accuracy : 0.9444
                 95% CI : (0.8134, 0.9932)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 1.728e-14

                  Kappa : 0.9167
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Iris-setosa Class: Iris-versicolor Class: Iris-virginica
Sensitivity                      1.0000                 0.9167                0.9167
Specificity                      1.0000                 0.9583                0.9583
Pos Pred Value                   1.0000                 0.9167                0.9167
Neg Pred Value                   1.0000                 0.9583                0.9583
Prevalence                       0.3333                 0.3333                0.3333
Detection Rate                   0.3333                 0.3056                0.3056
Detection Prevalence             0.3333                 0.3333                0.3333
Balanced Accuracy                1.0000                 0.9375                0.9375
```

We see that by using this algorithm, we misclassify two observations (one from the Versicolor specie and one from Virginica).

Support Vector Machines (SVM)

We also consider Support Vector Machines (SVM) with Radial Basis Function (RBF) Kernel. Basically, the SVM classifier maps the input space into a new space by a kernel transformation, and then finds the optimal separating hyperplane and the margin of separations in that space. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new observations [8].

For this model, the tuning parameters are the cost value (C and the radius of the RBF (sigma)). sigma is a parameter for the kernel function that can be used to expand/contract the distance function and C is the cost parameter that can be used as a regularization term that controls the complexity of the model. A high cost value C will force the SVM to create a complex enough prediction function to missclassify as few training points as possible, while a lower cost parameter will lead to a simpler prediction function [9]. In order to find the optimal values for sigma and C, we use bootstrap method.

```
Support Vector Machines with Radial Basis Function Kernel

114 samples
  4 predictor
  3 classes: 'Iris-setosa', 'Iris-versicolor', 'Iris-virginica'

No pre-processing
```

```
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 114, 114, 114, 114, 114, 114, ...
Resampling results across tuning parameters:

  C     Accuracy   Kappa
  0.25  0.9427309  0.9133450
  0.50  0.9510964  0.9259337
  1.00  0.9476927  0.9207770

Tuning parameter 'sigma' was held constant at a value of 0.5112317
Accuracy was used to select the optimal model using  the largest value.
The final values used for the model were sigma = 0.5112317 and C = 0.5.
```

The terminal output above shows that, based on accuracy, the optimal values are sigma = 0.5112317 and C = 0.5.

First, we can check how SVM performs on the training set. Below we can see the results from the terminal output.

```
Confusion Matrix and Statistics

                 Reference
Prediction        Iris-setosa Iris-versicolor Iris-virginica
  Iris-setosa              38               0              0
  Iris-versicolor           0              37              1
  Iris-virginica            0               1             37

Overall Statistics

               Accuracy : 0.9825
                 95% CI : (0.9381, 0.9979)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9737
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Iris-setosa Class: Iris-versicolor Class: Iris-virginica
Sensitivity                      1.0000                 0.9737                0.9737
Specificity                      1.0000                 0.9868                0.9868
Pos Pred Value                   1.0000                 0.9737                0.9737
Neg Pred Value                   1.0000                 0.9868                0.9868
Prevalence                       0.3333                 0.3333                0.3333
Detection Rate                   0.3333                 0.3246                0.3246
Detection Prevalence             0.3333                 0.3333                0.3333
Balanced Accuracy                1.0000                 0.9803                0.9803
```

Now, we can check how SVM performs on the testing set. Below we can see the results from the terminal output.

```
Confusion Matrix and Statistics

                 Reference
Prediction        Iris-setosa Iris-versicolor Iris-virginica
  Iris-setosa              12               0              0
  Iris-versicolor           0              11              1
  Iris-virginica            0               1             11

Overall Statistics
```

```
               Accuracy : 0.9444
                 95% CI : (0.8134, 0.9932)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 1.728e-14

                  Kappa : 0.9167
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Iris-setosa Class: Iris-versicolor Class: Iris-virginica
Sensitivity                      1.0000                 0.9167                0.9167
Specificity                      1.0000                 0.9583                0.9583
Pos Pred Value                   1.0000                 0.9167                0.9167
Neg Pred Value                   1.0000                 0.9583                0.9583
Prevalence                       0.3333                 0.3333                0.3333
Detection Rate                   0.3333                 0.3056                0.3056
Detection Prevalence             0.3333                 0.3333                0.3333
Balanced Accuracy                1.0000                 0.9375                0.9375
```

We see that by using SVM algorithm, we misclassify two observations (one from the Versicolor specie and one from Virginica).

Neural Network (NN)

A Neural Network (NN) is a computational model inspired by the way biological neural networks in the human brain process information. In other words, we can say that a neural network is a system composed of many simple processing elements which can acquire, store, and utilize experiential knowledge [10]. It consists of units (neurons), arranged in layers, which convert an input into some output. Each unit takes an input, applies a (often nonlinear) function to it and then passes the output on to the next layer [11].

For this model, the tuning parameters are size and decay. size is the number of units in hidden layer and decay is a regularization parameter to avoid over-fitting. In order to find the optimal values for size and decay, we use bootstrap method.

```
Neural Network

114 samples
  4 predictor
  3 classes: 'Iris-setosa', 'Iris-versicolor', 'Iris-virginica'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 114, 114, 114, 114, 114, 114, ...
Resampling results across tuning parameters:

  size  decay  Accuracy   Kappa
  1     0e+00  0.7425578  0.6251721
  1     1e-04  0.8929140  0.8385321
  1     1e-01  0.9630445  0.9440695
  3     0e+00  0.9082831  0.8633233
  3     1e-04  0.9645168  0.9463682
  3     1e-01  0.9782344  0.9672106
  5     0e+00  0.9485909  0.9225628
  5     1e-04  0.9642046  0.9459385
  5     1e-01  0.9792100  0.9686490

Accuracy was used to select the optimal model using  the largest value.
The final values used for the model were size = 5 and decay = 0.1.
```

The terminal output above shows that, based on accuracy, the optimal values are size = 5 and decay = 0.1.

First, we can check how NN performs on the training set. Below we can see the results from the terminal output.

```
Confusion Matrix and Statistics

                 Reference
Prediction        Iris-setosa Iris-versicolor Iris-virginica
  Iris-setosa              38               0              0
  Iris-versicolor           0              37              0
  Iris-virginica            0               1             38

Overall Statistics

               Accuracy : 0.9912
                 95% CI : (0.9521, 0.9998)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9868
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Iris-setosa Class: Iris-versicolor Class: Iris-virginica
Sensitivity                      1.0000                 0.9737                1.0000
Specificity                      1.0000                 1.0000                0.9868
Pos Pred Value                   1.0000                 1.0000                0.9744
Neg Pred Value                   1.0000                 0.9870                1.0000
Prevalence                       0.3333                 0.3333                0.3333
Detection Rate                   0.3333                 0.3246                0.3333
Detection Prevalence             0.3333                 0.3246                0.3421
Balanced Accuracy                1.0000                 0.9868                0.9934
```

Now, we can check how NN performs on the testing set. Below we can see the results from the terminal output.

```
Confusion Matrix and Statistics

                 Reference
Prediction        Iris-setosa Iris-versicolor Iris-virginica
  Iris-setosa              12               0              0
  Iris-versicolor           0              11              0
  Iris-virginica            0               1             12

Overall Statistics

               Accuracy : 0.9722
                 95% CI : (0.8547, 0.9993)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 4.864e-16

                  Kappa : 0.9583
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Iris-setosa Class: Iris-versicolor Class: Iris-virginica
Sensitivity                      1.0000                 0.9167                1.0000
Specificity                      1.0000                 1.0000                0.9583
Pos Pred Value                   1.0000                 1.0000                0.9231
Neg Pred Value                   1.0000                 0.9600                1.0000
Prevalence                       0.3333                 0.3333                0.3333
Detection Rate                   0.3333                 0.3056                0.3333
Detection Prevalence             0.3333                 0.3056                0.3611
Balanced Accuracy                1.0000                 0.9583                0.9792
```

We see that by using NN algorithm, we misclassify only one observation (from the Versicolor specie).

Summarizing the Models

We now have 8 models on the Iris dataset and accuracy estimations for each of them. As a final step, we can summarize the results by creating a list of the training set results for the developed models, and then select the most accurate one.

```
Call:
summary.resamples(object = results)

Models: DecisionTree, RandomForest, GBM, LDA, SVM, NeuralNetwork, kNN
Number of resamples: 25

Accuracy
                Min.   1st Qu.    Median      Mean   3rd Qu. Max. NA's
DecisionTree  0.9047619 0.9285714 0.9512195 0.9516993 0.9750000    1    0
RandomForest  0.9000000 0.9302326 0.9722222 0.9551032 0.9761905    1    0
GBM           0.9047619 0.9318182 0.9523810 0.9570530 0.9756098    1    0
LDA           0.9500000 0.9750000 0.9767442 0.9809617 1.0000000    1    0
SVM           0.8750000 0.9333333 0.9545455 0.9510964 0.9761905    1    0
NeuralNetwork 0.9268293 0.9565217 0.9761905 0.9792100 1.0000000    1    0
kNN           0.8780488 0.9523810 0.9761905 0.9713767 1.0000000    1    0

Kappa
                Min.   1st Qu.    Median      Mean   3rd Qu. Max. NA's
DecisionTree  0.8571429 0.8910035 0.9266547 0.9269501 0.9620853    1    0
RandomForest  0.8461538 0.8951220 0.9578947 0.9318577 0.9637306    1    0
GBM           0.8571429 0.8970252 0.9270833 0.9348769 0.9632945    1    0
LDA           0.9249531 0.9614272 0.9649266 0.9710940 1.0000000    1    0
SVM           0.8082454 0.8973384 0.9299674 0.9259337 0.9637306    1    0
NeuralNetwork 0.8905694 0.9348442 0.9637306 0.9686490 1.0000000    1    0
kNN           0.8189046 0.9284497 0.9635417 0.9567316 1.0000000    1    0
```

Moreover, we can create a plot of the model evaluation results and compare the spread and the mean accuracy of each model. There is a population of accuracy measures for each algorithm because each algorithm was evaluated 10 times (10-fold cross validation).

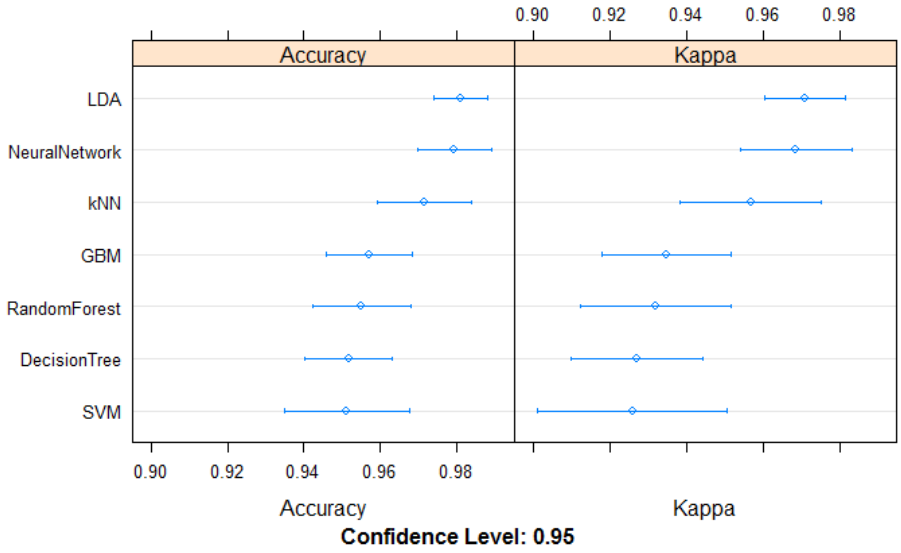

Figure 8: Evaluation results for all models.

We can see that the most accurate model in this case was LDA, which has Accuracy = 0.9809 and Kappa = 0.9710 on the training set and Accuracy = 0.9722 and Kappa = 0.9583 on the testing set. On the testing set, LDA misclassified only one observation: it predicted Virginica when it was actually Versicolor. Actually, this kind of misclassification is something that was expected from the exploratory analysis, since Virginica and Versicolor were the least distinguishable among the tree species.

References

[1] P-N Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining, Chapter 8. *Cluster Analysis: Basic Concepts and Algorithms*, Addison-Wesley (1st ed.), 2005

[2] Z-P Liu, Linear Discriminant Analysis, In Encyclopedia of Systems Biology, pp 1132–1133, Springer New York, 2013

[3] T. Mitchell, *Machine Learning*, McGraw Hill, 1997

[4] L. Breiman, "Random forests", Machine Learning, Vol. 45, No. 1, pp 5–32, 2001

[5] "Random forest", [Online]. Available: https://en.wikipedia.org/wiki/Random_forest, [Accessed: 07- Oct- 2017]

[6] T. Drabas, Practical Data Analysis Cookbook, Chapter 3. *Classification Techniques*, p 91, Packt Publishing, 2011

[7] M.H. Namaki, K. Sasani, Y. Wu and A.H. Gebremedhin, Performance Prediction for Graph Queries, In SIGMOD, 2017

[8] S. Wan and M.W. Mak, *Machine Learning for Protein Subcellular Localization Prediction*, p 167, De Gruyter, 2015

[9] A. Karatzoglou, D. Meyer and K. Hornik, Support Vector Machines in R, Journal of Statistical Software, Vol. 15, No. 9, 2006

[10] A. Pandya and R. Macy, *Pattern Recognition with Neural Networks in C++*, p 43, CRC Press and IEEE Press, 1996

[11] J. Brownlee, "Non-Linear Regression in R", 2014 [Online]. Available: https://machinelearningmastery.com/non-linear-regression-in-r/, [Accessed: 07- Oct- 2017]