

Data Exploration in R

Rémi FERRARI

The following documents the data exploration of a clean flower dataset called Iris. The project also takes a dive into statistical modelling. The programming language used in this project was the R language. Writing and computation of the code was realized on "rstudio.cloud".

Table of Contents

Importing Dataset	2
Summary Statistics.....	2
Visual insight	4
Classification Model Construction & Observation	7
Observing Model Decision Boundaries	13
Further Modelling	14
Optimization of "svmPoly" Model	20

Importing Dataset

Importing Iris dataset

```
> library(datasets)
> data("iris")
> |
```

Summary Statistics

Summary Statistics of Iris dataset

```
> head(iris, 5)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4          0.2  setosa
2          4.9         3.0          1.4          0.2  setosa
3          4.7         3.2          1.3          0.2  setosa
4          4.6         3.1          1.5          0.2  setosa
5          5.0         3.6          1.4          0.2  setosa

> tail(iris, 5)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
146          6.7         3.0          5.2          2.3 virginica
147          6.3         2.5          5.0          1.9 virginica
148          6.5         3.0          5.2          2.0 virginica
149          6.2         3.4          5.4          2.3 virginica
150          5.9         3.0          5.1          1.8 virginica

> summary(iris)
  Sepal.Length      Sepal.Width      Petal.Length      Petal.Width      Species
Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa :50
1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
Mean   :5.843   Mean   :3.057   Mean   :3.758   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

> library(skimr)
> skim(iris)
-- Data Summary --
Name      iris
Number of rows      150
Number of columns    5

Column type frequency:
  factor      1
  numeric     4

Group variables      None

-- Variable type: factor --
skim_variable n_missing complete_rate ordered n_unique top_counts
1 Species      0          1 FALSE          3 set: 50, ver: 50, vir: 50

-- Variable type: numeric --
skim_variable n_missing complete_rate mean      sd  p0 p25  p50 p75  p100 hist
1 Sepal.Length      0          1 5.84 0.828 4.3 5.1 5.8 6.4 7.9
2 Sepal.Width       0          1 3.06 0.436 2 2.8 3 3.3 4.4
3 Petal.Length      0          1 3.76 1.77 1 1.6 4.35 5.1 6.9
4 Petal.Width       0          1 1.20 0.762 0.1 0.3 1.3 1.8 2.5
```

Grouping by species

```
> iris %>%
+   dplyr::group_by(Species) %>%
+   skim()
```

```
— Data Summary —————
Name                               Values
Number of rows                    Piped data
Number of columns                  150
                                   5
```

```
Column type frequency:
  numeric                    4
```

```
Group variables                Species
```

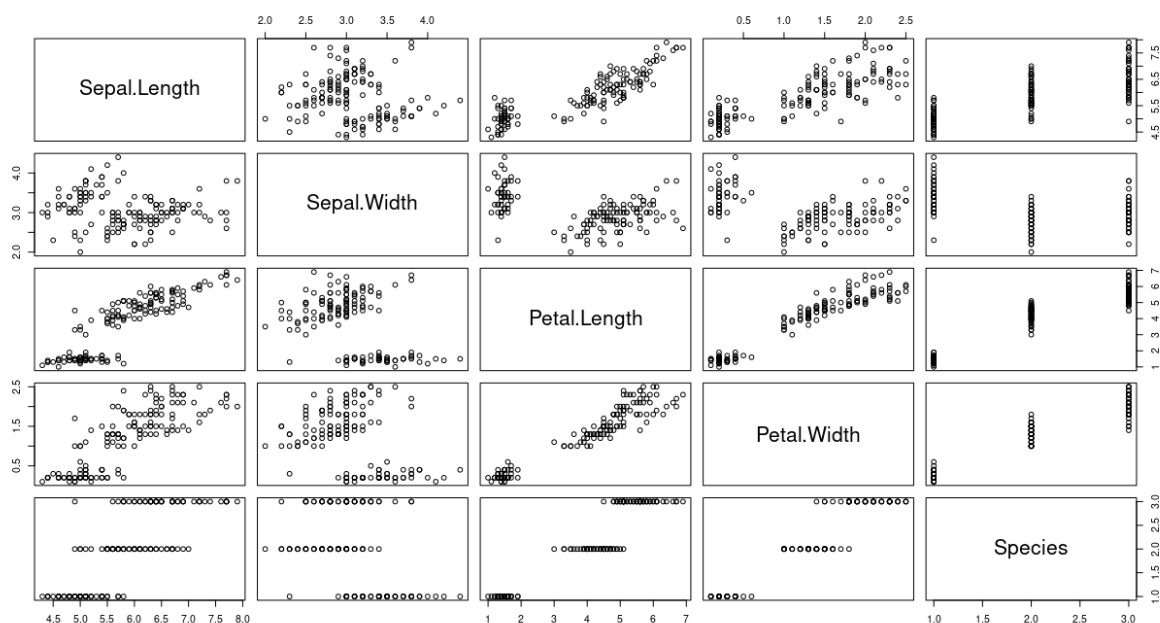
```
— Variable type: numeric —————
skim_variable Species    n_missing complete_rate mean    sd  p0  p25  p50  p75  p100
1 Sepal.Length setosa      0             1 5.01 0.352 4.3 4.8  5   5.2  5.8
2 Sepal.Length versicolor  0             1 5.94 0.516 4.9 5.6  5.9  6.3  7
3 Sepal.Length virginica   0             1 6.59 0.636 4.9 6.22 6.5  6.9  7.9
4 Sepal.Width  setosa      0             1 3.43 0.379 2.3 3.2  3.4  3.68 4.4
5 Sepal.Width  versicolor  0             1 2.77 0.314 2   2.52 2.8  3   3.4
6 Sepal.Width  virginica   0             1 2.97 0.322 2.2 2.8  3   3.18 3.8
7 Petal.Length setosa      0             1 1.46 0.174 1   1.4  1.5  1.58 1.9
8 Petal.Length versicolor  0             1 4.26 0.470 3   4   4.35 4.6  5.1
9 Petal.Length virginica   0             1 5.55 0.552 4.5 5.1  5.55 5.88 6.9
10 Petal.Width setosa      0             1 0.246 0.105 0.1 0.2  0.2  0.3  0.6
11 Petal.Width versicolor  0             1 1.33 0.198 1   1.2  1.3  1.5  1.8
12 Petal.Width virginica   0             1 2.03 0.275 1.4 1.8  2   2.3  2.5
```



Visual insight

Panel Plot

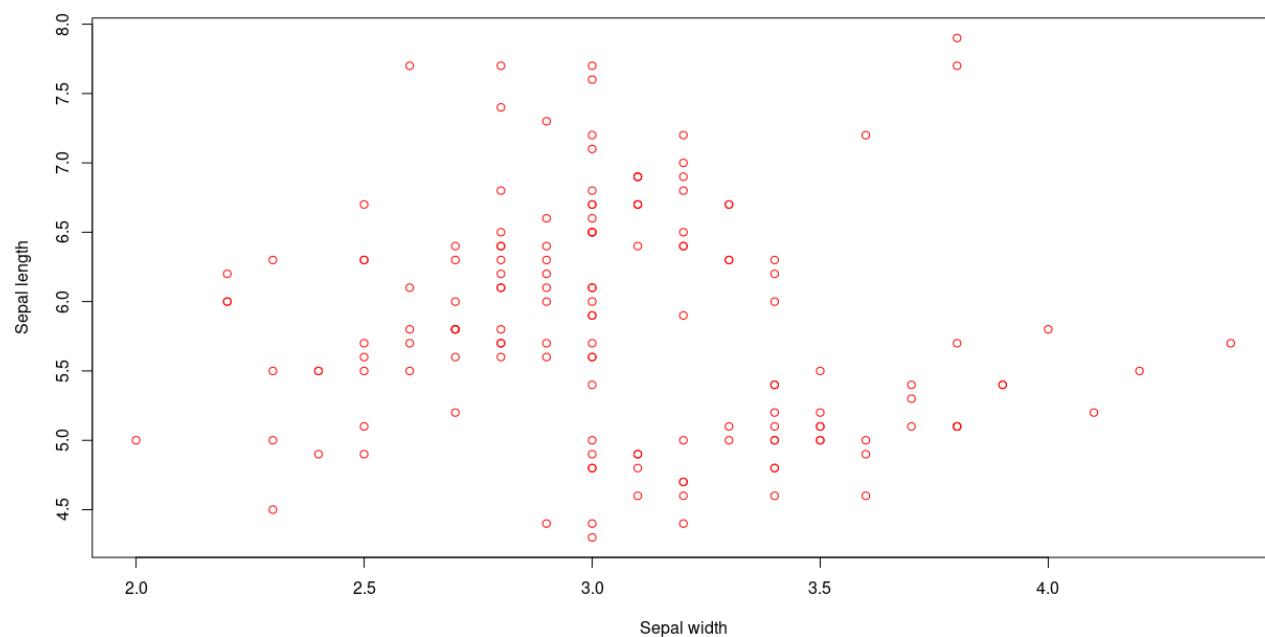
```
> plot(iris)
```

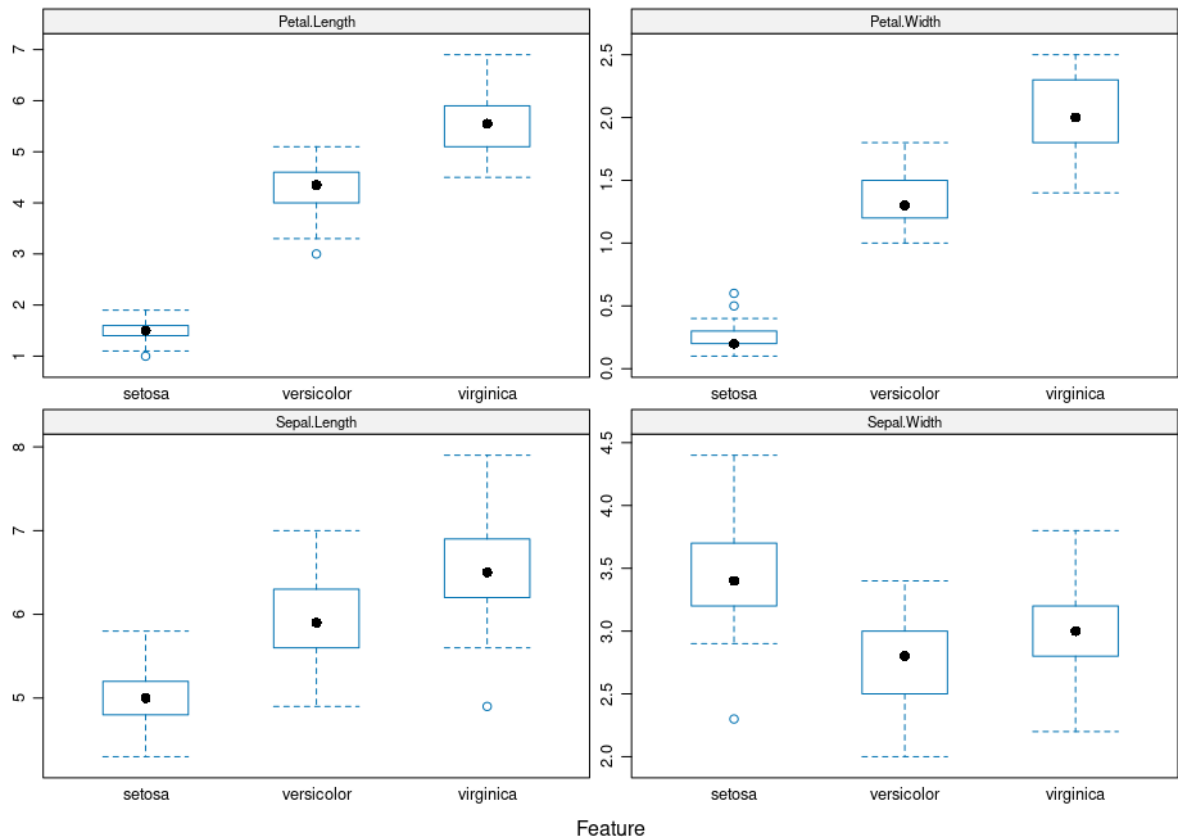


Scatter Plot

```
> plot(iris$Sepal.Width, iris$Sepal.Length)
>
> plot(iris$Sepal.Width, iris$Sepal.Length, col = "red")
>
> plot(iris$Sepal.Width, iris$Sepal.Length, col = "red",
+       xlab = "Sepal width", ylab = "Sepal length")
```

(For output, see below)





Observations:

We can observe that Virginica species is generally lengthier in sepal and petal length. It is generally wider in petal width. Throughout petal length, petal width, and sepal length we can observe several common characteristics: Virginica species is, in general, the larger species followed up by the versicolor species. Same can be said when observing the range of the species throughout the different metrics. These characteristics are not shown in sepal width with the setosa species showcasing similar range to the other species and a generally greater value.

Classification Model Construction & Observation

Setting random seed number, separating and observing training and testing data

```
set.seed(100)
```

```
TrainingIndex <- createDataPartition(iris$Species, p=0.8, list = FALSE)
TrainingSet <- iris[TrainingIndex,]
TestingSet <- iris[-TrainingIndex,]

> View(TrainingSet)
> View(TestingSet)
```

Testing and training set can be observed below with a 4:1 training set to testing set ratio. Two bottom figures showcase training (left) and testing data (right)

Environment	History	Connections	Tutorial
<div> Import Dataset <div>420 MiB</div> </div> <div> <div>List</div> <div>Refresh</div> </div>			
R - Global Environment			
Data			
iris	150 obs. of 5 variables		
TestingSet	30 obs. of 5 variables		
TrainingIndex	int [1:120, 1] 1 2 4 6 7 8 9 10 11 12 ...		
TrainingSet	120 obs. of 5 variables		

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species		Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa	3	4.7	3.2	1.3	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa	5	5.0	3.6	1.4	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa	17	5.4	3.9	1.3	0.4	setosa
6	5.4	3.9	1.7	0.4	setosa	24	5.1	3.3	1.7	0.5	setosa
7	4.6	3.4	1.4	0.3	setosa	28	5.2	3.5	1.5	0.2	setosa
8	5.0	3.4	1.5	0.2	setosa	32	5.4	3.4	1.5	0.4	setosa
9	4.4	2.9	1.4	0.2	setosa	39	4.4	3.0	1.3	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa	45	5.1	3.8	1.9	0.4	setosa
11	5.4	3.7	1.5	0.2	setosa	46	4.8	3.0	1.4	0.3	setosa
12	4.8	3.4	1.6	0.2	setosa	50	5.0	3.3	1.4	0.2	setosa
13	4.8	3.0	1.4	0.1	setosa	52	6.4	3.2	4.5	1.5	versicolor
14	4.3	3.0	1.1	0.1	setosa	56	5.7	2.8	4.5	1.3	versicolor
15	5.8	4.0	1.2	0.2	setosa	61	5.0	2.0	3.5	1.0	versicolor
16	5.7	4.4	1.5	0.4	setosa	63	6.0	2.2	4.0	1.0	versicolor
18	5.1	3.5	1.4	0.3	setosa	67	5.6	3.0	4.5	1.5	versicolor
19	5.7	3.8	1.7	0.3	setosa	80	5.7	2.6	3.5	1.0	versicolor
20	5.1	3.8	1.5	0.3	setosa	81	5.5	2.4	3.8	1.1	versicolor
21	5.4	3.4	1.7	0.2	setosa	84	6.0	2.7	5.1	1.6	versicolor
22	5.1	3.7	1.5	0.4	setosa	90	5.5	2.5	4.0	1.3	versicolor
23	4.6	3.6	1.0	0.2	setosa	95	5.6	2.7	4.2	1.3	versicolor
25	4.8	3.4	1.9	0.2	setosa	107	4.9	2.5	4.5	1.7	virginica
26	5.0	3.0	1.6	0.2	setosa	109	6.7	2.5	5.8	1.8	virginica
27	5.0	3.4	1.6	0.4	setosa	113	6.8	3.0	5.5	2.1	virginica
29	5.2	3.4	1.4	0.2	setosa	124	6.3	2.7	4.9	1.8	virginica
30	4.7	3.2	1.6	0.2	setosa	125	6.7	3.3	5.7	2.1	virginica
31	4.8	3.1	1.6	0.2	setosa	129	6.4	2.8	5.6	2.1	virginica
33	5.2	4.1	1.5	0.1	setosa	132	7.9	3.8	6.4	2.0	virginica
34	5.5	4.2	1.4	0.2	setosa	136	7.7	3.0	6.1	2.3	virginica

Realizing and observing scatterplots for training and testing sets in the purpose of gaining insight on training vs testing sets distribution differences. (Not all scatterplots are shown)

```
> plot(TrainingSet$Sepal.Width, TrainingSet$Sepal.Length, xlab = "Sepal Width", ylab = "Sepal Length")
> plot(TestingSet$Sepal.Width, TestingSet$Sepal.Length, xlab = "Sepal Width", ylab = "Sepal Length")
```

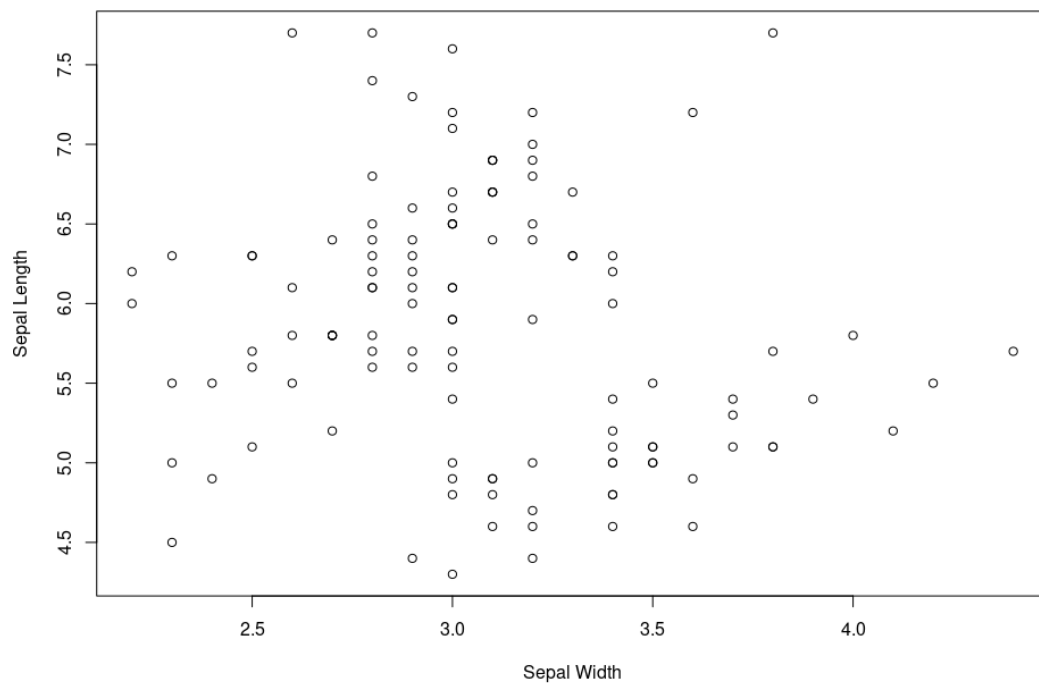


Figure 1: Training Set: Sepal Width vs Sepal Length

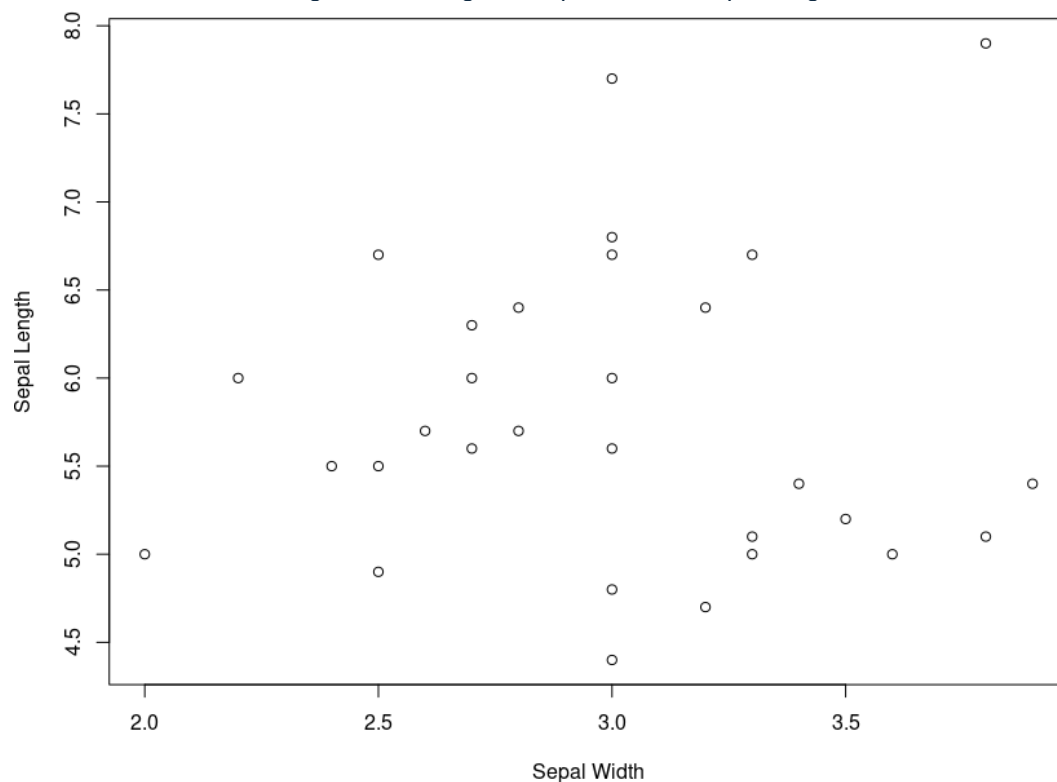


Figure 2: Testing Set: Sepal Width vs Sepal Length

Training Model

A training model is a machine learning model that is created by learning patterns from a labeled dataset (training data). The purpose of a training model is to adjust its parameters to minimize errors and improve performance, enabling it to make accurate predictions on new, unseen data.

```
> Model <- train(Species ~ ., data = TrainingSet,
+               method = "svmPoly",
+               na.action = na.omit,
+               preProcess=c("scale","center"),
+               trControl= trainControl(method="none"),
+               tuneGrid = data.frame(degree=1,scale=1,C=1)
+ )
```

Cross-Valdiation Model

A cross-validation model is a technique used to evaluate the generalizability of a machine learning model by splitting the data into multiple subsets (folds). The purpose of cross-validation is to assess how well the model will perform on independent datasets, helping to avoid overfitting and ensure robust performance across different data samples.

```
> Model.cv <- train(Species ~ ., data = TrainingSet,
+                  method = "svmPoly",
+                  na.action = na.omit,
+                  preProcess=c("scale","center"),
+                  trControl= trainControl(method="cv", number=10),
+                  tuneGrid = data.frame(degree=1,scale=1,C=1)
+ )
```

Applying Model for Prediction and Observation of Results

```
> Model.training <-predict(Model, TrainingSet)
> Model.testing <-predict(Model, TestingSet)
> Model.cv <-predict(Model.cv, TrainingSet)
```

Values	
Model.cv	Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...
Model.testing	Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...
Model.training	Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...

(See output below)

```
> print(Model.training.confusion)
Confusion Matrix and Statistics
```

	Reference		
Prediction	setosa	versicolor	virginica
setosa	40	0	0
versicolor	0	40	1
virginica	0	0	39

Overall Statistics

Accuracy : 0.9917
 95% CI : (0.9544, 0.9998)
 No Information Rate : 0.3333
 P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.9875

Mcnemar's Test P-Value : NA

Statistics by Class:

	Class: setosa	Class: versicolor	Class: virginica
Sensitivity	1.0000	1.0000	0.9750
Specificity	1.0000	0.9875	1.0000
Pos Pred Value	1.0000	0.9756	1.0000
Neg Pred Value	1.0000	1.0000	0.9877
Prevalence	0.3333	0.3333	0.3333
Detection Rate	0.3333	0.3333	0.3250
Detection Prevalence	0.3333	0.3417	0.3250
Balanced Accuracy	1.0000	0.9938	0.9875

Overall, the model performs exceptionally well on the training set, with high accuracy, sensitivity, specificity, and balanced accuracy for all classes. The results indicate that the model has learned to classify the iris species very effectively, with minimal errors.

```
> print(Model.testing.confusion)
Confusion Matrix and Statistics
```

	Reference		
Prediction	setosa	versicolor	virginica
setosa	10	0	0
versicolor	0	9	0
virginica	0	1	10

Overall Statistics

Accuracy : 0.9667
 95% CI : (0.8278, 0.9992)
 No Information Rate : 0.3333
 P-Value [Acc > NIR] : 2.963e-13

Kappa : 0.95

Mcnemar's Test P-Value : NA

Statistics by Class:

	Class: setosa	Class: versicolor	Class: virginica
Sensitivity	1.0000	0.9000	1.0000
Specificity	1.0000	1.0000	0.9500
Pos Pred Value	1.0000	1.0000	0.9091
Neg Pred Value	1.0000	0.9524	1.0000
Prevalence	0.3333	0.3333	0.3333
Detection Rate	0.3333	0.3000	0.3333
Detection Prevalence	0.3333	0.3000	0.3667
Balanced Accuracy	1.0000	0.9500	0.9750

The model performs excellently on the testing set, with high accuracy and robust classification performance across all classes.

```
> print(Model.cv.confusion)
```

Confusion Matrix and Statistics

	Reference		
Prediction	setosa	versicolor	virginica
setosa	40	0	0
versicolor	0	40	1
virginica	0	0	39

Overall Statistics

Accuracy : 0.9917
 95% CI : (0.9544, 0.9998)
 No Information Rate : 0.3333
 P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.9875

McNemar's Test P-Value : NA

Statistics by Class:

	Class: setosa	Class: versicolor	Class: virginica
Sensitivity	1.0000	1.0000	0.9750
Specificity	1.0000	0.9875	1.0000
Pos Pred Value	1.0000	0.9756	1.0000
Neg Pred Value	1.0000	1.0000	0.9877
Prevalence	0.3333	0.3333	0.3333
Detection Rate	0.3333	0.3333	0.3250
Detection Prevalence	0.3333	0.3417	0.3250
Balanced Accuracy	1.0000	0.9938	0.9875

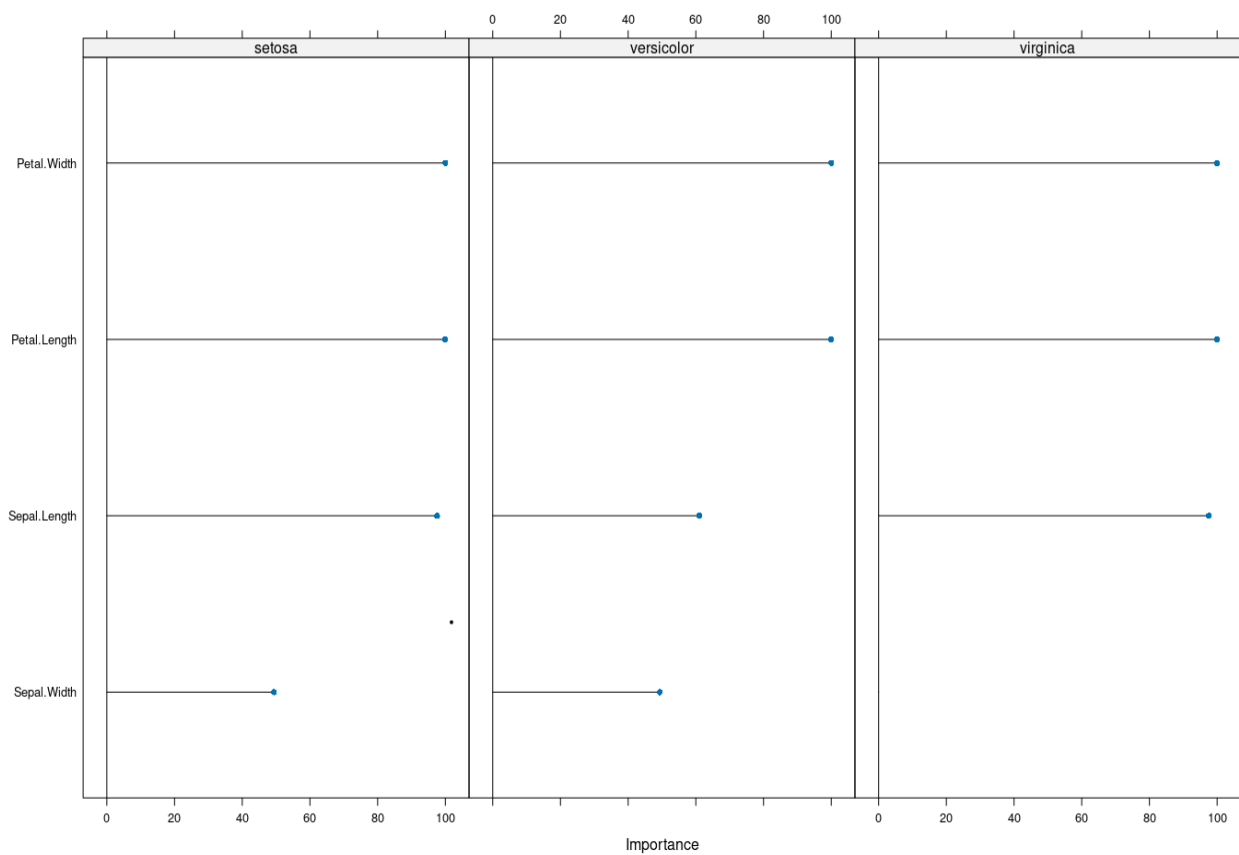
The cross-validation results show that the model has high accuracy and performs excellently across all classes, demonstrating robust and reliable classification performance.

Feature Importance

```
> Importance <- varImp(Model)
> plot(Importance)
> plot(Importance)
```

(see below for output)

Graph



```
> importance <- varImp(Model, scale = FALSE)
```

```
> print(importance)
```

```
ROC curve variable importance
```

```
variables are sorted by maximum importance across the classes
```

	setosa	versicolor	virginica
Petal.Length	1.0000	1.0000	1.0000
Petal.Width	1.0000	1.0000	1.0000
Sepal.Length	0.9959	0.9350	0.9959
Sepal.Width	0.9156	0.9156	0.8334

Observation:

Petal Length and Petal Width are the most important features for predicting the species in the Iris dataset, based on the ROC curve analysis.

Sepal Length is also important but slightly less influential.

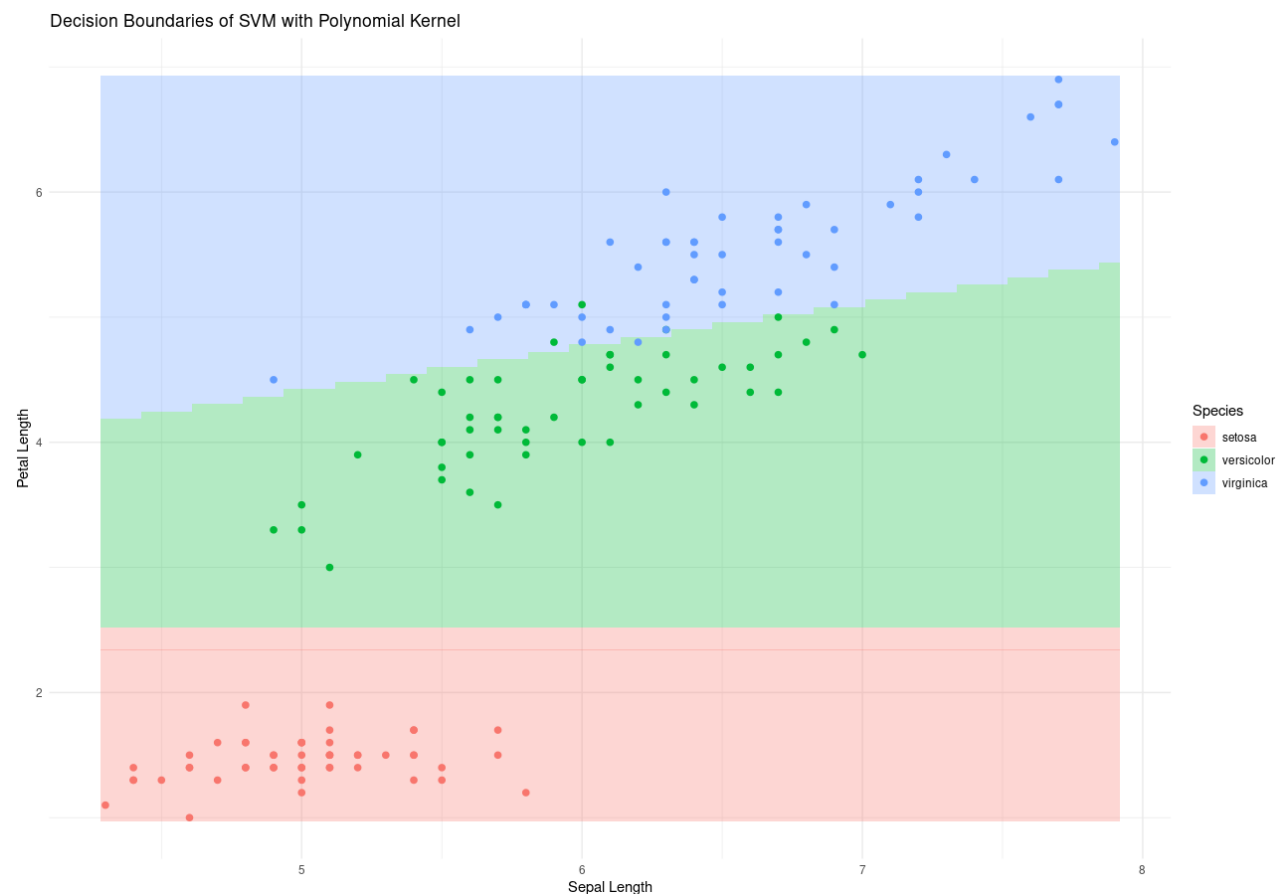
Sepal Width is the least important feature, although it still plays a role in the model's predictions.

These results suggest that the petal measurements (length and width) are much more crucial for distinguishing between the species compared to the sepal measurements.

Observing Model Decision Boundaries

Model Decision Boundaries

Please note that the decision boundaries graphics below are based upon a model utilizing the same method but restricted to two features.



Observations:

Model predicts that the Setosa species area remains between a petal length of 4cm and a sepal length of 8cm (this last number is most probably way off and showcases limits of the model and data). Versicor species ranges in petal length from 4 to 5.5cm as a linear function to Sepa length which ranges from 4 to 8cm. Setosa species ranges in sepal length from 4 to 8cm and petal length from 4 and above. All flowers of the virginica family where correctly identified. Several flowers from the Versicolor and Virginica where incorrectly identified. Overall, the majority of flowers where correctly identified. Please note that further information must be observed in order to correctly assess the model's predictive capacity.

Further Modelling

“svmLinear” method

```
> Model <- train(Species ~ ., data = TrainingSet,
+               method = "svmLinear",
+               na.action = na.omit,
+               preProcess = c("scale", "center"),
+               trControl = trainControl(method = "none"),
+               tuneGrid = data.frame(C = 1),
+               probability = TRUE
+ )
>
> Model.cv <- train(Species ~ ., data = TrainingSet,
+                  method = "svmLinear",
+                  na.action = na.omit,
+                  preProcess = c("scale", "center"),
+                  trControl = trainControl(method = "cv", number = 10),
+                  tuneGrid = data.frame(C = 1),
+                  probability = TRUE
+ )
>
```

Confusion Matrix and Statistics

	Reference		
Prediction	setosa	versicolor	virginica
setosa	40	0	0
versicolor	0	40	1
virginica	0	0	39

Overall Statistics

```
Accuracy : 0.9917
95% CI : (0.9544, 0.9998)
No Information Rate : 0.3333
P-Value [Acc > NIR] : < 2.2e-16
```

```
Kappa : 0.9875
```

```
McNemar's Test P-Value : NA
```

Statistics by Class:

	Class: setosa	Class: versicolor	Class: virginica
Sensitivity	1.0000	1.0000	0.9750
Specificity	1.0000	0.9875	1.0000
Pos Pred Value	1.0000	0.9756	1.0000
Neg Pred Value	1.0000	1.0000	0.9877
Prevalence	0.3333	0.3333	0.3333
Detection Rate	0.3333	0.3333	0.3250
Detection Prevalence	0.3333	0.3417	0.3250
Balanced Accuracy	1.0000	0.9938	0.9875

Confusion Matrix and Statistics

Prediction	Reference		
	setosa	versicolor	virginica
setosa	10	0	0
versicolor	0	9	0
virginica	0	1	10

Overall Statistics

Accuracy : 0.9667
 95% CI : (0.8278, 0.9992)
 No Information Rate : 0.3333
 P-Value [Acc > NIR] : 2.963e-13

Kappa : 0.95

Mcnemar's Test P-Value : NA

Statistics by Class:

	Class: setosa	Class: versicolor	Class: virginica
Sensitivity	1.0000	0.9000	1.0000
Specificity	1.0000	1.0000	0.9500
Pos Pred Value	1.0000	1.0000	0.9091
Neg Pred Value	1.0000	0.9524	1.0000
Prevalence	0.3333	0.3333	0.3333
Detection Rate	0.3333	0.3000	0.3333
Detection Prevalence	0.3333	0.3000	0.3667
Balanced Accuracy	1.0000	0.9500	0.9750

> print(Model.cv.confusion)

Confusion Matrix and Statistics

Prediction	Reference		
	setosa	versicolor	virginica
setosa	40	0	0
versicolor	0	40	1
virginica	0	0	39

Overall Statistics

Accuracy : 0.9917
 95% CI : (0.9544, 0.9998)
 No Information Rate : 0.3333
 P-Value [Acc > NIR] : < 2.2e-16

Kappa : 0.9875

Mcnemar's Test P-Value : NA

Statistics by Class:

	Class: setosa	Class: versicolor	Class: virginica
Sensitivity	1.0000	1.0000	0.9750
Specificity	1.0000	0.9875	1.0000
Pos Pred Value	1.0000	0.9756	1.0000
Neg Pred Value	1.0000	1.0000	0.9877
Prevalence	0.3333	0.3333	0.3333
Detection Rate	0.3333	0.3333	0.3250
Detection Prevalence	0.3333	0.3417	0.3250
Balanced Accuracy	1.0000	0.9938	0.9875

Observations:

Interestingly, the svmLinear model gets the same results as svmPoly. This is mainly due to the smaller size of the dataset. Additionally, however, the svmPoly model used a degree value of 1 effectively making it linear.

“svmPoly” method (degree of 3)

```
> Model <- train(Species ~ ., data = TrainingSet,
+               method = "svmPoly",
+               na.action = na.omit,
+               preProcess = c("scale", "center"),
+               trControl = trainControl(method = "none"),
+               tuneGrid = data.frame(degree = 3, scale = 1, C = 1),
+               probability = TRUE
+ )
>
> Model.cv <- train(Species ~ ., data = TrainingSet,
+                  method = "svmPoly",
+                  na.action = na.omit,
+                  preProcess = c("scale", "center"),
+                  trControl = trainControl(method = "cv", number = 10),
+                  tuneGrid = data.frame(degree = 3, scale = 1, C = 1),
+                  probability = TRUE
+ )
```

```
> print(Model.training.confusion)
```

Confusion Matrix and Statistics

	Reference		
Prediction	setosa	versicolor	virginica
setosa	40	0	0
versicolor	0	40	0
virginica	0	0	40

Overall Statistics

```
Accuracy : 1
95% CI : (0.9697, 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: setosa	Class: versicolor	Class: 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


```
> print(Model.testing.confusion)
```

Confusion Matrix and Statistics

	Reference		
Prediction	setosa	versicolor	virginica
setosa	10	0	0
versicolor	0	9	0
virginica	0	1	10

Overall Statistics

Accuracy : 0.9667
 95% CI : (0.8278, 0.9992)
 No Information Rate : 0.3333
 P-Value [Acc > NIR] : 2.963e-13

Kappa : 0.95

McNemar's Test P-Value : NA

Statistics by Class:

	Class: setosa	Class: versicolor	Class: virginica
Sensitivity	1.0000	0.9000	1.0000
Specificity	1.0000	1.0000	0.9500
Pos Pred Value	1.0000	1.0000	0.9091
Neg Pred Value	1.0000	0.9524	1.0000
Prevalence	0.3333	0.3333	0.3333
Detection Rate	0.3333	0.3000	0.3333
Detection Prevalence	0.3333	0.3000	0.3667
Balanced Accuracy	1.0000	0.9500	0.9750

```
> print(Model.cv.confusion)
```

Confusion Matrix and Statistics

	Reference		
Prediction	setosa	versicolor	virginica
setosa	40	0	0
versicolor	0	40	0
virginica	0	0	40

Overall Statistics

Accuracy : 1
 95% CI : (0.9697, 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: setosa	Class: versicolor	Class: 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

```
>
```

As we can observe this svmPoly model differs from the linear svmPoly and svmLinear models. All training data points are correctly identified. Accuracy is equal to 1 versus 0.9917. It performs to the same degree in testing data. Interestingly, cross-validation model also shows improvements.

“svmRadial” method

```
> Model <- train(Species ~ ., data = TrainingSet,
+               method = "svmRadial",
+               na.action = na.omit,
+               preProcess = c("scale", "center"),
+               trControl = trainControl(method = "none"),
+               tuneGrid = data.frame(sigma = 0.1, C = 1),
+               probability = TRUE
+ )
>
> Model.cv <- train(Species ~ ., data = TrainingSet,
+                  method = "svmRadial",
+                  na.action = na.omit,
+                  preProcess = c("scale", "center"),
+                  trControl = trainControl(method = "cv", number = 10),
+                  tuneGrid = data.frame(sigma = 0.1, C = 1),
+                  probability = TRUE
+ )
> print(Model.training.confusion)
Confusion Matrix and Statistics
```

	Reference		
Prediction	setosa	versicolor	virginica
setosa	40	0	0
versicolor	0	39	1
virginica	0	1	39

Overall Statistics

```
Accuracy : 0.9833
95% CI : (0.9411, 0.998)
No Information Rate : 0.3333
P-Value [Acc > NIR] : < 2.2e-16
```

```
Kappa : 0.975
```

```
McNemar's Test P-Value : NA
```

Statistics by Class:

	Class: setosa	Class: versicolor	Class: virginica
Sensitivity	1.0000	0.9750	0.9750
Specificity	1.0000	0.9875	0.9875
Pos Pred Value	1.0000	0.9750	0.9750
Neg Pred Value	1.0000	0.9875	0.9875
Prevalence	0.3333	0.3333	0.3333
Detection Rate	0.3333	0.3250	0.3250
Detection Prevalence	0.3333	0.3333	0.3333
Balanced Accuracy	1.0000	0.9812	0.9812

```
> print(Model.testing.confusion)
Confusion Matrix and Statistics
```

	Reference		
Prediction	setosa	versicolor	virginica
setosa	10	0	0
versicolor	0	9	1
virginica	0	1	9

Overall Statistics

```
Accuracy : 0.9333
95% CI : (0.7793, 0.9918)
No Information Rate : 0.3333
P-Value [Acc > NIR] : 8.747e-12
```

```
Kappa : 0.9
```

```
McNemar's Test P-Value : NA
```

Statistics by Class:

	Class: setosa	Class: versicolor	Class: virginica
Sensitivity	1.0000	0.9000	0.9000
Specificity	1.0000	0.9500	0.9500
Pos Pred Value	1.0000	0.9000	0.9000
Neg Pred Value	1.0000	0.9500	0.9500
Prevalence	0.3333	0.3333	0.3333
Detection Rate	0.3333	0.3000	0.3000
Detection Prevalence	0.3333	0.3333	0.3333
Balanced Accuracy	1.0000	0.9250	0.9250

```
> print(Model.cv.confusion)
Confusion Matrix and Statistics
```

	Reference		
Prediction	setosa	versicolor	virginica
setosa	40	0	0
versicolor	0	39	1
virginica	0	1	39

Overall Statistics

```
Accuracy : 0.9833
95% CI : (0.9411, 0.998)
No Information Rate : 0.3333
P-Value [Acc > NIR] : < 2.2e-16
```

```
Kappa : 0.975
```

```
McNemar's Test P-Value : NA
```

Statistics by Class:

	Class: setosa	Class: versicolor	Class: virginica
Sensitivity	1.0000	0.9750	0.9750
Specificity	1.0000	0.9875	0.9875
Pos Pred Value	1.0000	0.9750	0.9750
Neg Pred Value	1.0000	0.9875	0.9875
Prevalence	0.3333	0.3333	0.3333
Detection Rate	0.3333	0.3250	0.3250
Detection Prevalence	0.3333	0.3333	0.3333
Balanced Accuracy	1.0000	0.9812	0.9812

```
> |
```

So far, this model has the least performance with accuracy levels ranging from 0.9333-0.9833

Optimization of “svmPoly” Model

svmPoly model was optimized for cost and degree parameters.

Grid Search optimization method

```
> library(caret)
>
>
> tuneGrid <- expand.grid(
+   degree = c(2, 3, 4, 5),
+   scale = c(0.1, 1, 10, 100),
+   C = c(0.1, 1, 10, 100)
+ )
>
>
> print(paste("Number of parameter combinations (Grid Search):", nrow(tuneGrid)))
[1] "Number of parameter combinations (Grid Search): 64"
>
>
> set.seed(100)
> Model_poly <- train(
+   Species ~ .,
+   data = TrainingSet,
+   method = "svmPoly",
+   na.action = na.omit,
+   preProcess = c("scale", "center"),
+   trControl = trainControl(method = "cv", number = 10),
+   tuneGrid = tuneGrid,
+   probability = TRUE
+ )
>
>
> print(paste("Number of parameter combinations tested:", nrow(Model_poly$results)))
[1] "Number of parameter combinations tested: 64"
>
> print(Model_poly$bestTune)
      degree scale    C
13         2  100.0  0.1
```

Here are results:

```
> print(best_combinations)
      degree scale    C Accuracy Kappa AccuracySD   KappaSD
13         2  100.0  0.1  0.9833333 0.975 0.03513642 0.05270463
14         2  100.0  1.0  0.9833333 0.975 0.03513642 0.05270463
15         2  100.0 10.0  0.9833333 0.975 0.03513642 0.05270463
16         2  100.0 100.0 0.9833333 0.975 0.03513642 0.05270463
18         3    0.1   1.0  0.9833333 0.975 0.03513642 0.05270463
```

Testing of result 18:

(see below)

```
> print(Model.cv.confusion)
Confusion Matrix and Statistics
```

	Reference		
Prediction	setosa	versicolor	virginica
setosa	40	0	0
versicolor	0	39	1
virginica	0	1	39

Overall Statistics

```
Accuracy : 0.9833
95% CI : (0.9411, 0.998)
No Information Rate : 0.3333
P-Value [Acc > NIR] : < 2.2e-16
```

```
Kappa : 0.975
```

Please note: An anomaly occurred in which results from the optimization process did not match the output of certain models with specified hyperparameters despite having the same hyperparameters. Source of anomaly could not be found