Export an R Model to Python-PySpark

Basic example of exporting a R Model to a Python-PySpark instance .

When more classes are to be learnt, one speaks of a multi-class problem, such as annotation of a new Iris example as being from the setosa, versicolor or virginica species. In these cases, the output is a single label (of one of the anticipated classes).

Source – https://lgatto.github.io/IntroMachineLearningWithR/supervised-learning.html

```
#install.packages("ggplot2")
#install.packages("ggfortify")
#install.packages("caret")
#install.packages("class")
# install.packages("gridExtra")
# install.packages("GGally")
# install.packages("RGraphics")
# install.packages("gmodels")
# install.packages("tibble")
#install.packages("plotly")
#install.packages("e1071")
install.packages("webshot")
```

```
## Installing package into '/home/dhankar/R/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
webshot::install_phantomjs()
```

```
## It seems that the version of `phantomjs` installed is greater than or equal to the requested version
```

```
#

library("ggplot2")
library("ggfortify")
library("caret")
```

```
## Loading required package: lattice
```

```
library("class")
library("gridExtra")
library("GGally")
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library("RGraphics")
```

```
## Loading required package: grid
```

```r
library("gmodels")
require("tibble")
```

```
## Loading required package: tibble
```

```r
library(ggplot2)
library(ggfortify)
library(caret)
library(e1071)
#library(here)
library(tibble)
#library(class)
library(gridExtra)

library(plotly)
```

```
##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout
```

```r
library(GGally)
library(plotly)
library(gmodels)
```

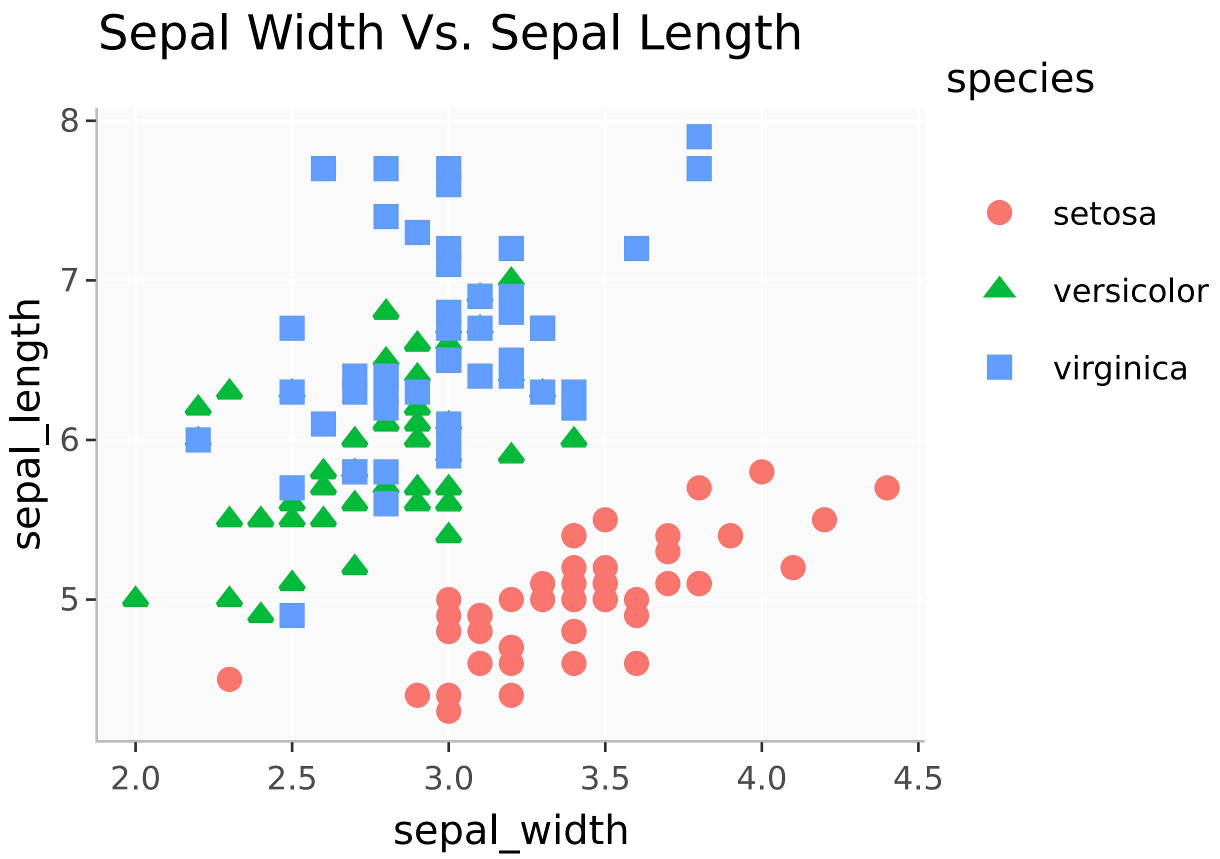
```r
attach(iris)
data(iris)
iris_tb <- as_tibble(iris)
head(iris_tb)
```

```
## # A tibble: 6 x 5
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##          <dbl>       <dbl>        <dbl>       <dbl> <fct>
## 1          5.1         3.5          1.4         0.2 setosa
## 2          4.9         3            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           3.6          1.4         0.2 setosa
## 6          5.4         3.9          1.7         0.4 setosa
```

```r
colnames(iris_tb) <- c('sepal_length', 'sepal_width',
                       'petal_length', 'petal_width',
                       'species')
```

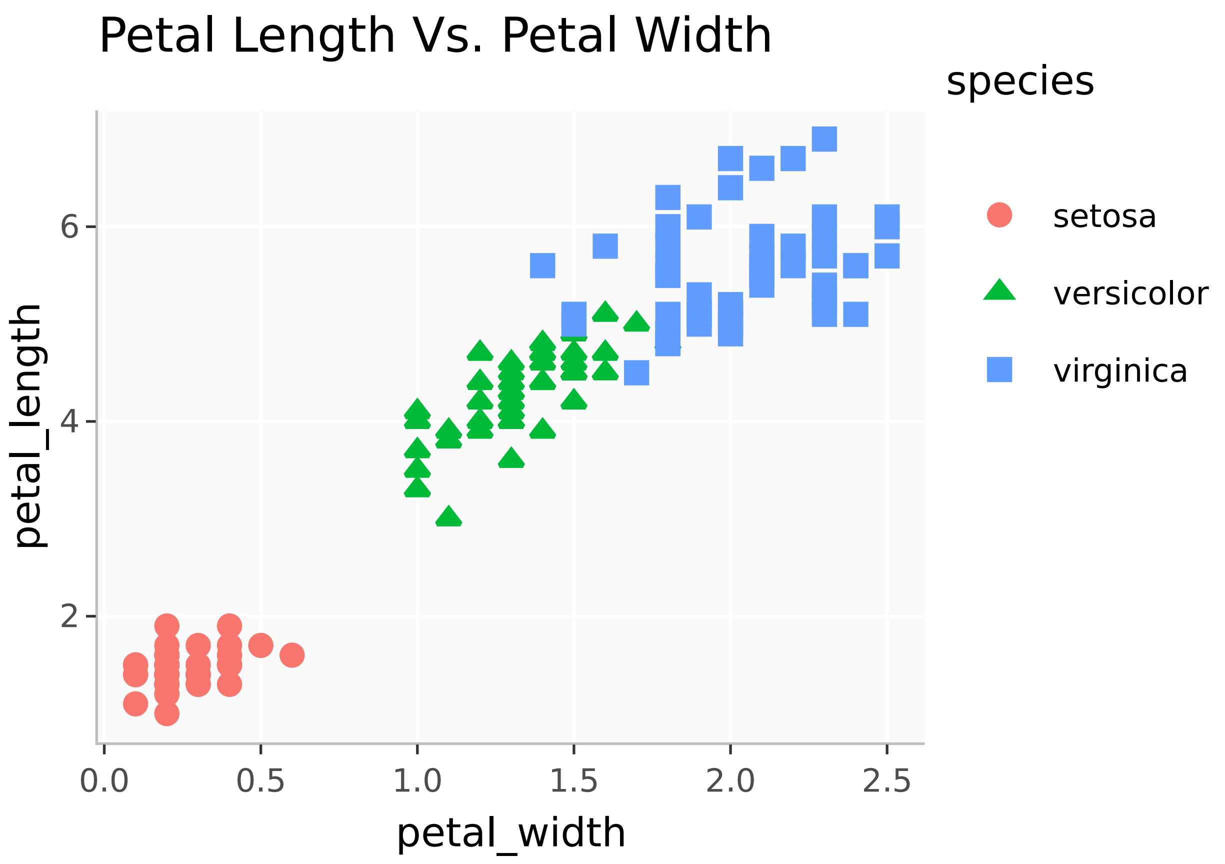
```r
gg1<-ggplot(iris_tb,
            aes(x=sepal_width,y=sepal_length,
                shape=species,
                color=species)) +
  theme(panel.background = element_rect(fill = "gray98"),
        axis.line   = element_line(colour="black"),
        axis.line.x = element_line(colour="gray"),
        axis.line.y = element_line(colour="gray")) +
  geom_point(size=2) +
  labs(title = "Sepal Width Vs. Sepal Length")

ggplotly(gg1)
```

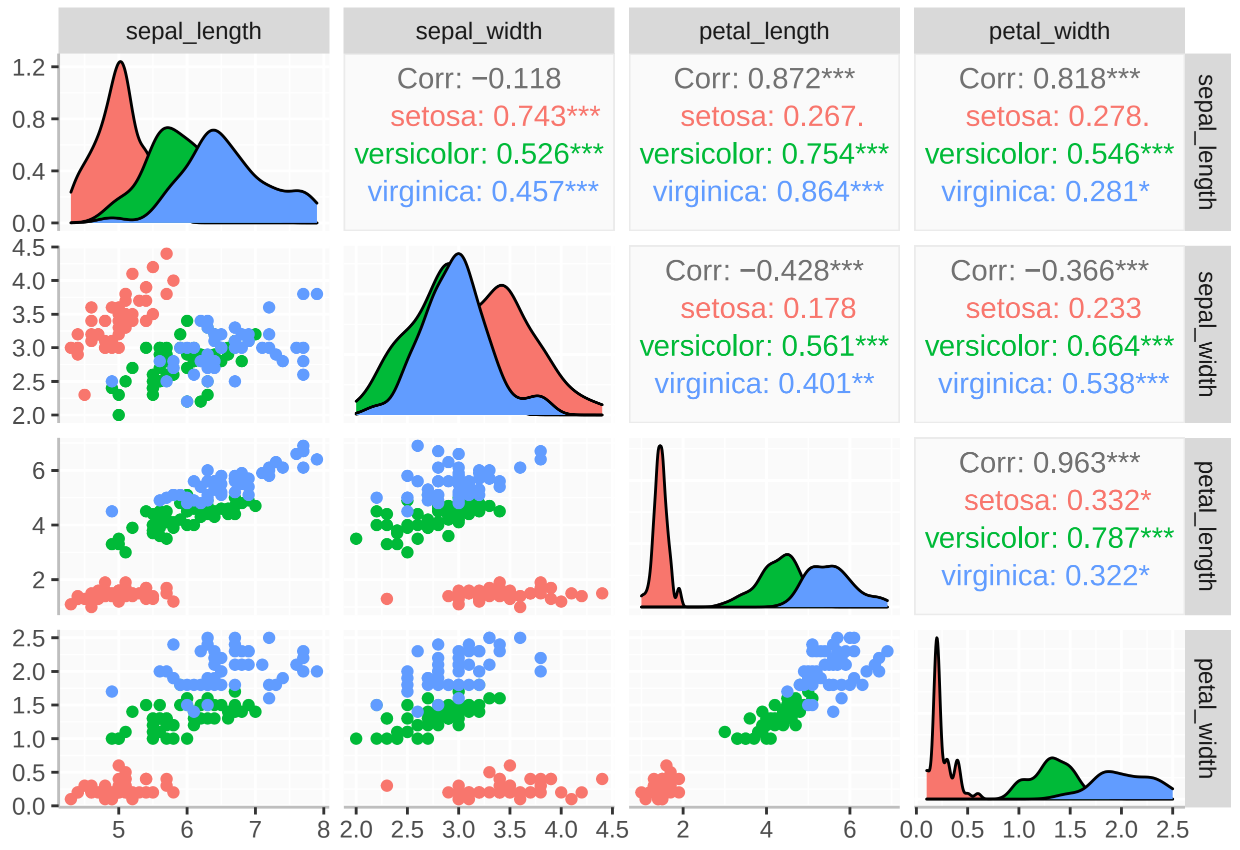


```
gg2<-ggplot(iris_tb,
            aes(x=petal_width,y=petal_length,
                shape=species,
                color=species)) +
  theme(panel.background = element_rect(fill = "gray98"),
        axis.line   = element_line(colour="black"),
        axis.line.x = element_line(colour="gray"),
        axis.line.y = element_line(colour="gray")) +
  geom_point(size=2) +
  labs(title = "Petal Length Vs. Petal Width")

ggplotly(gg2)
```



```
pairs <- ggpairs(iris_tb,
                 mapping=aes(color=species),
                 columns=1:4) +
  theme(panel.background = element_rect(fill = "gray98"),
        axis.line   = element_line(colour="black"),
        axis.line.x = element_line(colour="gray"),
        axis.line.y = element_line(colour="gray"))
pairs
```

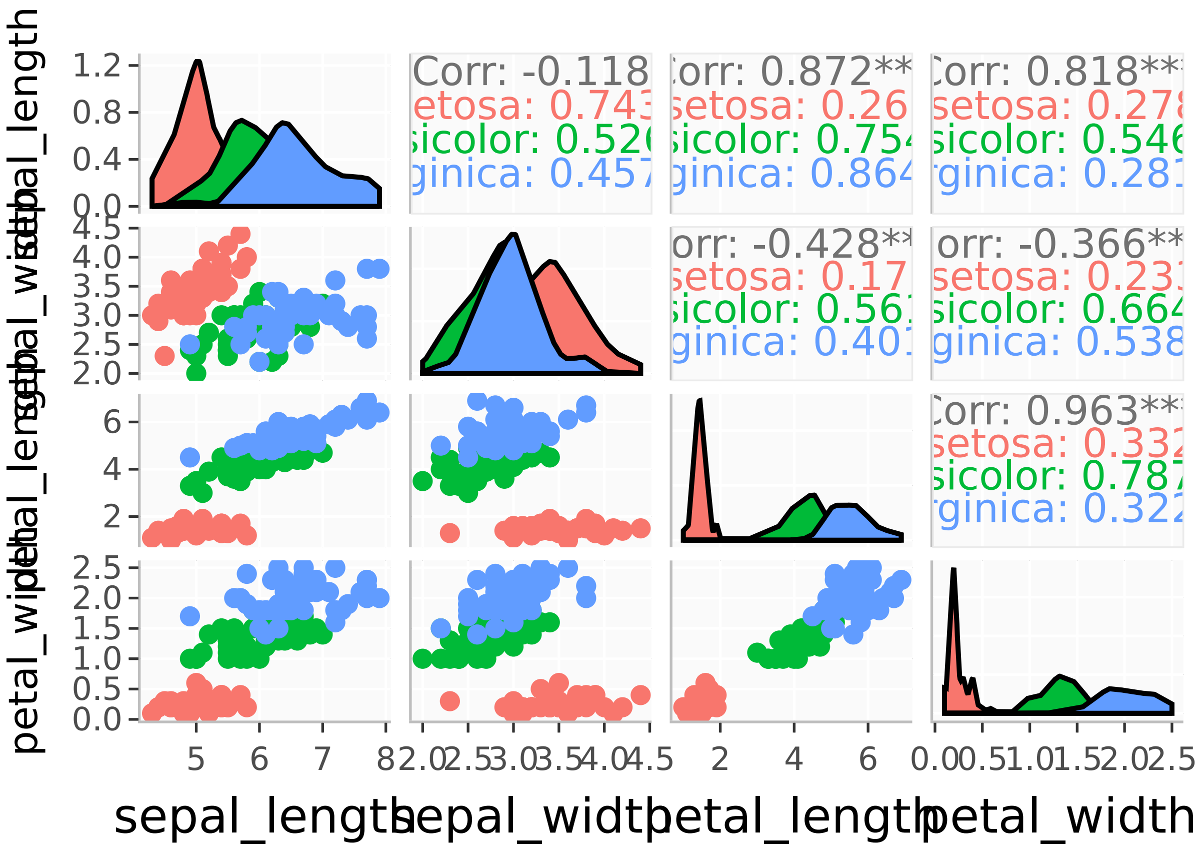


```r
ggplotly(pairs) %>%
  layout(showlegend = FALSE)
```

```
## Warning: Can only have one: highlight

## Warning: Can only have one: highlight

## Warning: Can only have one: highlight
```



```
## GGAlly Source -- https://github.com/rstats-gsoc/gsoc2016/wiki/ggduo:-pairs-plots-for-multiple-regres
```

```
# MODEL ESTIMATION
 # Creating training/test set
# Source -- https://github.com/jungsoonw/Practical_Machine_Learning_Assignment/wiki
# Package -- E1071-- Further Reading -- https://cran.r-project.org/web/packages/e1071/e1071.pdf
#

set.seed(88)
trainIndex <- createDataPartition(iris_tb$species,
                                  p = .8,
                                  list = FALSE,
                                  times = 1)
# Creating 80 20 split
training_set <- iris_tb[ trainIndex,]

test_set  <- iris_tb[-trainIndex,]

# USING CARET PACKAGE TO ESTIMATE OPTIMAL K

fit <- train(species ~ .,
             data = training_set,
             method = "knn")

# Here we output the results from fit!
fit
```

```
## k-Nearest Neighbors
##
## 120 samples
##   4 predictor
##   3 classes: 'setosa', 'versicolor', 'virginica'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 120, 120, 120, 120, 120, 120, ...
## Resampling results across tuning parameters:
##
##   k  Accuracy   Kappa
##   5  0.9561614  0.9337935
##   7  0.9629487  0.9441134
##   9  0.9666055  0.9496128
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 9.
```

```r
#
fit$results
```

```
##   k  Accuracy     Kappa AccuracySD    KappaSD
## 1 5 0.9561614 0.9337935 0.02654774 0.04002242
## 2 7 0.9629487 0.9441134 0.02729764 0.04105867
## 3 9 0.9666055 0.9496128 0.02063019 0.03102241
```

```r
#
class(fit) # [1] "train"         "train.formula"
```

```
## [1] "train"         "train.formula"
```

```r
typeof(fit) # List
```

```
## [1] "list"
```

```r
# PREDICTION RESULTS
# Predict
predict_test_set <- predict(fit,
                            newdata = test_set)
class(predict_test_set)
```

```
## [1] "factor"
```

```r
#
CrossTable(x = test_set$species,
           y = predict_test_set,
           prop.chisq=FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  30
##
##
##                  | predict_test_set
## test_set$species |     setosa | versicolor |  virginica |  Row Total |
## -----------------|------------|------------|------------|------------|
##           setosa |         10 |          0 |          0 |         10 |
##                  |      1.000 |      0.000 |      0.000 |      0.333 |
##                  |      1.000 |      0.000 |      0.000 |            |
##                  |      0.333 |      0.000 |      0.000 |            |
## -----------------|------------|------------|------------|------------|
##       versicolor |          0 |          9 |          1 |         10 |
##                  |      0.000 |      0.900 |      0.100 |      0.333 |
##                  |      0.000 |      1.000 |      0.091 |            |
##                  |      0.000 |      0.300 |      0.033 |            |
## -----------------|------------|------------|------------|------------|
##        virginica |          0 |          0 |         10 |         10 |
##                  |      0.000 |      0.000 |      1.000 |      0.333 |
##                  |      0.000 |      0.000 |      0.909 |            |
##                  |      0.000 |      0.000 |      0.333 |            |
## -----------------|------------|------------|------------|------------|
##     Column Total |         10 |          9 |         11 |         30 |
##                  |      0.333 |      0.300 |      0.367 |            |
## -----------------|------------|------------|------------|------------|
##
##
```

```r
# WE CAN ESTIMATE OUR TEST ERROR RATE AS FOLLOWS FROM OUR TABLE:
# > table(test_set$species, predict_test_set)
# predict_test_set
# setosa versicolor virginica
# setosa         10          0         0
# versicolor      0          8         2
# virginica       0          0        10

# Total Number of observations is 30
# Total Number of observations predicted incorrectly is 2

print(round(1 - fit$results$Accuracy[1], 4))
```

```
## [1] 0.0438
```

```r
# WE RECEIVE A TEST ERROR RATE OF 0.0334
```