Iris Classification Capstone Project

Luka Vujeva

6/17/2019

Table of Contents

Introduction

The goal of this project is to classify Iris flowers based on their epal length and width, and their petal length and width. This was done using a variety of machine learning algorithms, namely cross validation, linear discriminant analysis, k nearest neighbours, classification and regression trees. In the end, the most accurate algorithm will be used to classify the Irises. The dataset was provided by Kaggle, and the link to it is https://www.kaggle.com/uciml/iris/downloads/iris-species.zip/2 , but it can also be found in my git repository.

Data Cleaning

First we start by loading the caret library and importing the .csv file that the data is in.

```
library(caret)

## Warning: package 'caret' was built under R version 3.5.3

## Loading required package: lattice

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.5.3

df <- read.csv("Iris.csv")
```

To prepare the data for use in our case study, we need to split it into a train set and a validation set. To do this we will take 80% of the data as the training set, and use the remaining 20% as the validation set.

```
train <- createDataPartition(df$Species, p=.8, list=F)
test <- df[-train,]
```

Data Exploration

Here we will look at the basic attributes of the data set.

```
#next check the dimensions of the dataset
dim(df)

## [1] 150   6

#next we go through the types of attributes
sapply(df, class)

##            Id SepalLengthCm  SepalWidthCm PetalLengthCm  PetalWidthCm
##     "integer"     "numeric"     "numeric"     "numeric"     "numeric"
##       Species
##      "factor"

head(df)

##   Id SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm     Species
## 1  1           5.1          3.5           1.4          0.2 Iris-setosa
## 2  2           4.9          3.0           1.4          0.2 Iris-setosa
## 3  3           4.7          3.2           1.3          0.2 Iris-setosa
## 4  4           4.6          3.1           1.5          0.2 Iris-setosa
## 5  5           5.0          3.6           1.4          0.2 Iris-setosa
## 6  6           5.4          3.9           1.7          0.4 Iris-setosa

levels(df$Species)

## [1] "Iris-setosa"     "Iris-versicolor" "Iris-virginica"

summary(df)

##        Id         SepalLengthCm    SepalWidthCm   PetalLengthCm
##  Min.   :  1.00   Min.   :4.300   Min.   :2.000   Min.   :1.000
##  1st Qu.: 38.25   1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600
##  Median : 75.50   Median :5.800   Median :3.000   Median :4.350
##  Mean   : 75.50   Mean   :5.843   Mean   :3.054   Mean   :3.759
##  3rd Qu.:112.75   3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100
##  Max.   :150.00   Max.   :7.900   Max.   :4.400   Max.   :6.900
##   PetalWidthCm              Species
##  Min.   :0.100   Iris-setosa    :50
##  1st Qu.:0.300   Iris-versicolor:50
##  Median :1.300   Iris-virginica :50
##  Mean   :1.199
##  3rd Qu.:1.800
##  Max.   :2.500
```

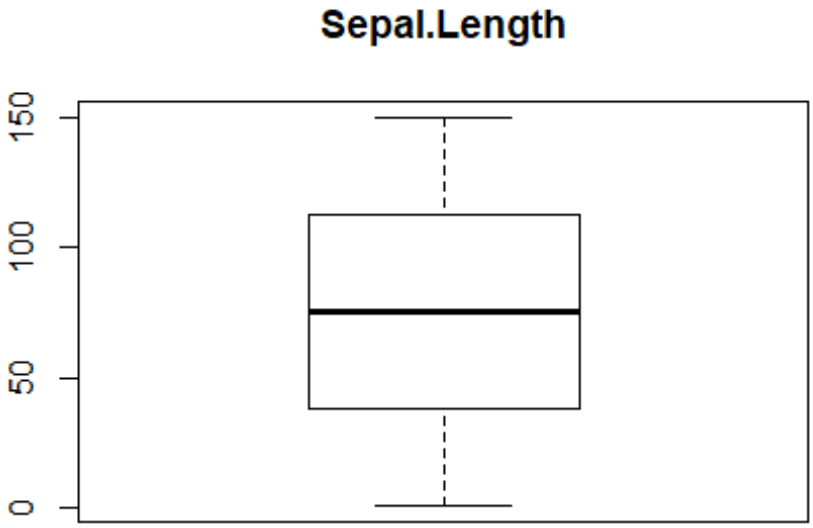
As we can see, the dataset contains 6 columns: Id, SepalLengthCm, SepalWidthCm, PetalWidthCm, PetalLengthCm and Species. The data set also contains 150 columns.

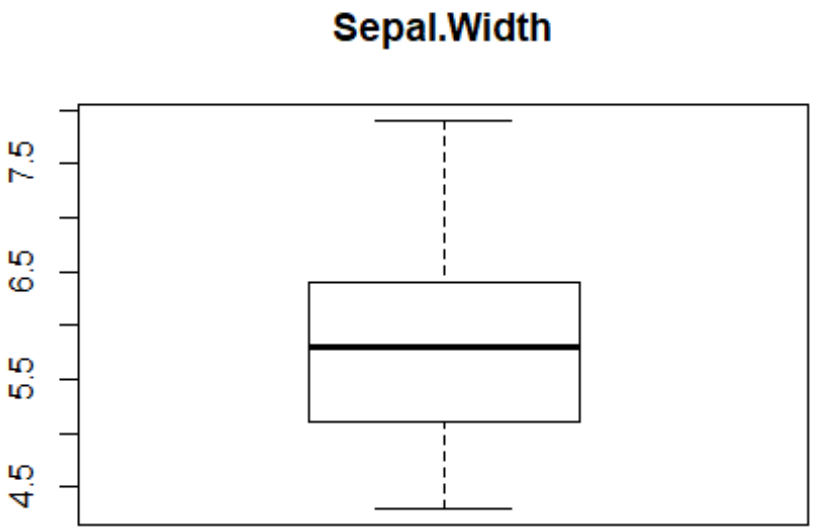
Next we split our data into x and y values.

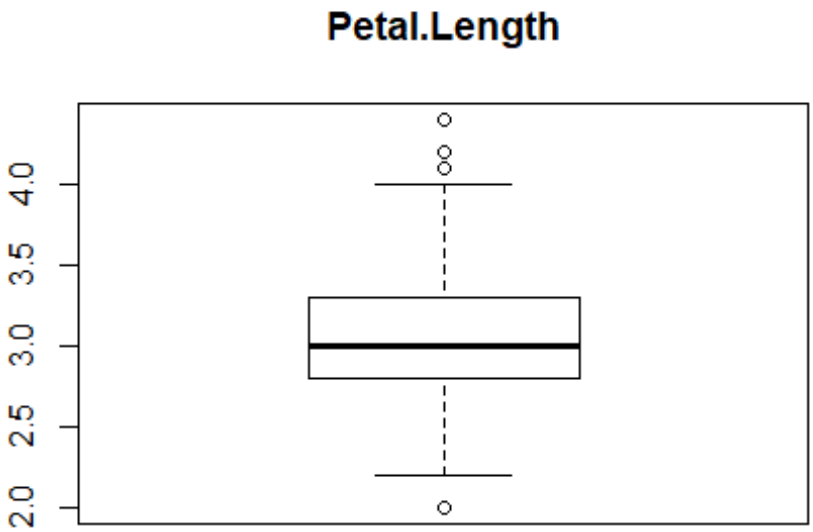
```
x <- df[,1:4]
y <- df[,5]

par(mfrow=c(1,4))
```

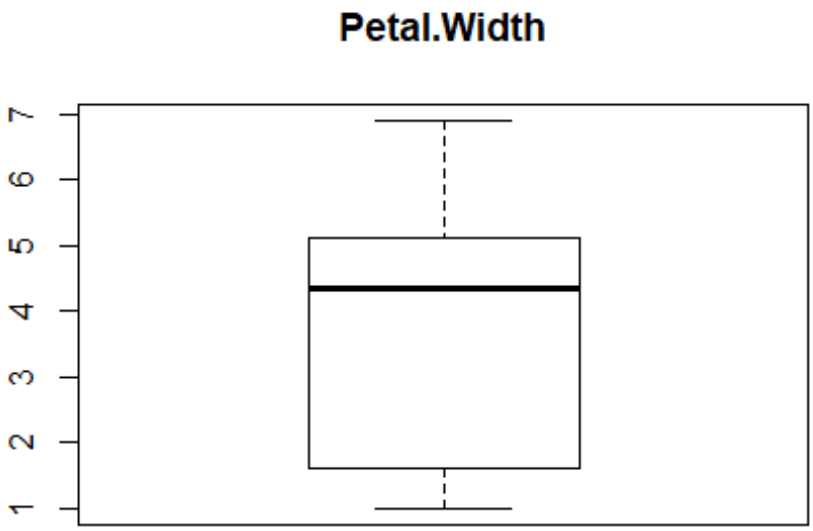
Next we will plot the data as a box charts to illustrate the range of the data

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


Sepal.Length
150
100
50
0


Sepal.Width
7.5
6.5
5.5
4.5


Petal.Length
2.0
2.5
3.0
3.5
4.0


Petal.Width
1
2
3
4
5
6
7

MODELING

In this section we will try a variety of machine learning algorithms to see which produces the most accurate resluts.

```
#we first try cross validation
control <- trainControl(method="cv", number=10)
metric = "Accuracy"
```

Accuracy the number of correctly predicted instances in the dataset given as a percentage

```
#Next we try Linear Discriminant Analysis (LDA)
set.seed(1)
fit.lda <- train(Species~., data=df, method="lda", metric=metric,
trControl=control)

#Next we can try the K Nearest Neighbours Method
set.seed(7)
fit.knn <- train(Species~., data=df, method="knn", metric=metric,
trControl=control)

# Finally we try Classification and Regression Trees
set.seed(7)
fit.cart <- train(Species~., data=df, method="rpart", metric=metric,
trControl=control)
```

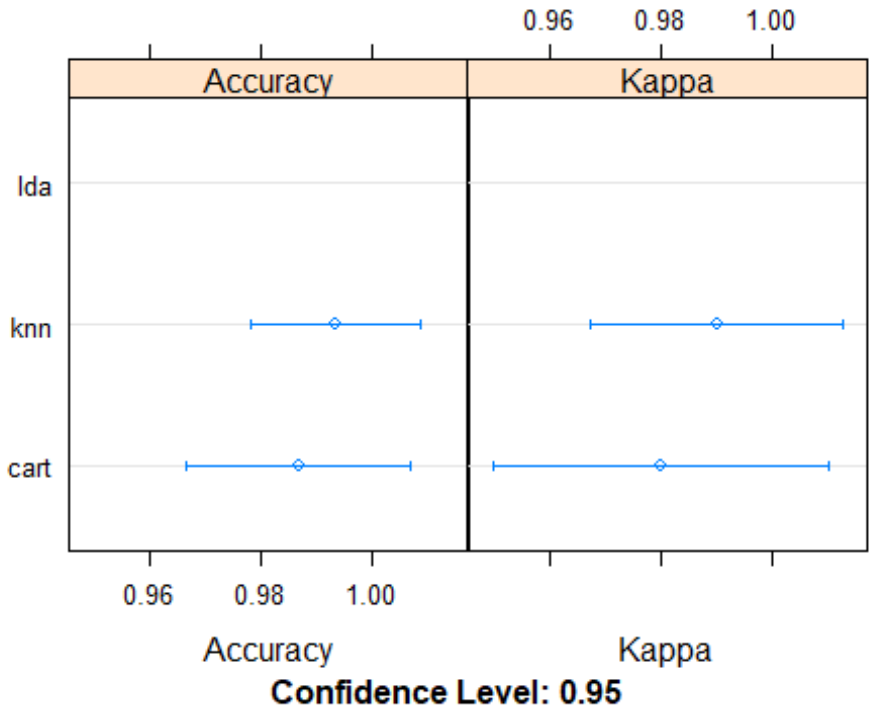
Results

Now that we have tried the various algorithms, we can put it all together and decide on the best one for our use case.

```
# Now we can put the accuracy of models in a table and determine which has
the highest accuracy
results <- resamples(list(lda=fit.lda, cart=fit.cart, knn=fit.knn))
summary(results)
```

```
##
## Call:
## summary.resamples(object = results)
##
## Models: lda, cart, knn
## Number of resamples: 10
##
## Accuracy
##           Min. 1st Qu. Median      Mean 3rd Qu. Max. NA's
## lda  1.0000000       1      1 1.0000000       1    1    0
## cart 0.9333333       1      1 0.9866667       1    1    0
## knn  0.9333333       1      1 0.9933333       1    1    0
##
## Kappa
```

```
##       Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
## lda   1.0       1      1 1.00       1    1    0
## cart  0.9       1      1 0.98       1    1    0
## knn   0.9       1      1 0.99       1    1    0

#next we plot the results in a dot plot
dotplot(results)
```




```
print(fit.lda)

## Linear Discriminant Analysis
##
## 150 samples
##   5 predictor
##   3 classes: 'Iris-setosa', 'Iris-versicolor', 'Iris-virginica'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 135, 135, 135, 135, 135, 135, ...
## Resampling results:
##
##   Accuracy  Kappa
##   1         1
```

As you can see in the summary and the dot plot above, though all of the methods were relatively accurate, the Linear Discriminant Analysis, or LDA, method was the most

accurate in classifying the Iris flowers, with the k nearest neighbours method coming in second, and the classification and regression trees method coming in third.

Finally, given that the LDA method is the most accurate, we can proceed in using it against our validation set, and looking at the results in a confusion matrix.

```
predictions <- predict(fit.lda, test)
confusionMatrix(predictions, test$Species)
```

```
## Confusion Matrix and Statistics
##
##                 Reference
## Prediction        Iris-setosa Iris-versicolor Iris-virginica
##   Iris-setosa              10               0              0
##   Iris-versicolor           0              10              0
##   Iris-virginica            0               0             10
##
## Overall Statistics
##
##                Accuracy : 1
##                  95% CI : (0.8843, 1)
##     No Information Rate : 0.3333
##     P-Value [Acc > NIR] : 4.857e-15
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Iris-setosa Class: Iris-versicolor
## Sensitivity                      1.0000                 1.0000
## Specificity                      1.0000                 1.0000
## Pos Pred Value                   1.0000                 1.0000
## Neg Pred Value                   1.0000                 1.0000
## Prevalence                       0.3333                 0.3333
## Detection Rate                   0.3333                 0.3333
## Detection Prevalence             0.3333                 0.3333
## Balanced Accuracy                1.0000                 1.0000
##                      Class: Iris-virginica
## Sensitivity                         1.0000
## Specificity                         1.0000
## Pos Pred Value                      1.0000
## Neg Pred Value                      1.0000
## Prevalence                          0.3333
## Detection Rate                      0.3333
## Detection Prevalence                0.3333
## Balanced Accuracy                   1.0000
```

Conclusion

Therefore, we can conclude that the methof of Linear Discriminant Analysis, or LDA, is the most accurate prediction method given our use of test and validation sets. It boasted an accuracy of 100% with a p-value of only 4.857e-15.