KNN Algorithem in Iris Dataset

Ladan Foroughi

Contents

List of Figures

List of Tables

Introduction

The Iris dataset or Fisher's Iris data set is a multivariate data set. This dataset consist of five attributes - sepal length, sepal width, petal length, petal width and species. The algorithm is used for prediction of species is KNN algorithm.

Loading Data

The data is uploading from < https://www.kaggle.com/uciml/iris?select=Iris.csv>. All the packaged that used in this work is downloading from http://cran.us.r-project.org

```
if(!require(pacman))install.packages("pacman")
pacman::p_load(
  tidyverse,
  dplyr,
  ggplot,
  caret,
  magnittr,
  pacman,
  GGally,
  knitr,
  parallel,
  rattel,
  tictoc,
  gridExtra,
  kableExtra,
  readr,
  purrr,
  randomForest,
  pROC,
  fastDummies,
  rpart.plot,
  data.table,
  reshape2,
  graphics,
  corrplot,
  latexpdf,
  ReporteRs,
  tinytex,
  latexdiffr,
  latex2exp,
  class,
)

temp <- tempfile()
url <- "https://www.kaggle.com/uciml/iris"
download.file(url, temp)
rawdata <- fread("iris.csv", header=TRUE)
```

```
unlink(temp)
iris <- rename(rawdata)
rm(rawdata,temp,url)
iris <- iris[,-1]
iris <- iris %>% rename('Petal Length'= PetalLengthCm,
                'Petal Width' = PetalWidthCm,
                'Sepal Length'= SepalLengthCm,
                'Sepal Width' = SepalWidthCm) %>%
  mutate(Species = fct_recode(Species,
                              'setosa' ='Iris-setosa',
                              'versicolor' = 'Iris-versicolor',
                              'virginica' = 'Iris-virginica'))
```

Before we started to analyze the data we need to know the information of dataset.

First rows

```
# First rows
kable(head(iris),
      "pandoc",
      caption = "The first six rows of data set",
      align = "c",
      font_size = 5)
```

Table 1: The first six rows of data set

Sepal Length	Sepal Width	Petal Length	Petal Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

Last rows

```
# Last rows
kable(tail(iris),
      "pandoc",
      caption = "The last six rows of data set",
      align = "c",
      font_size = 5)
```

Table 2: The last six rows of data set

Sepal Length	Sepal Width	Petal Length	Petal Width	Species
6.7	3.3	5.7	2.5	virginica
6.7	3.0	5.2	2.3	virginica
6.3	2.5	5.0	1.9	virginica
6.5	3.0	5.2	2.0	virginica
6.2	3.4	5.4	2.3	virginica
5.9	3.0	5.1	1.8	virginica

Summary

```
# Summary
kable(summary(iris),
      "pandoc",
      caption = "The summary of data set",
      align = "c",
      font_size = 5)
```

Table 3: The summary of data set

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 :3.054	Mean :3.759	Mean :1.199	NA
3rd Qu.:6.400	3rd Qu.:3.300	3rd Qu.:5.100	3rd Qu.:1.800	NA
Max. :7.900	Max. :4.400	Max. :6.900	Max. :2.500	NA

Structure

```
# Structure
kable(str(iris),
      "pandoc",
      caption = "The structure of data set",
      align = "c",
      font_size = 5)
```

```
## Classes 'data.table' and 'data.frame':   150 obs. of  5 variables:
##  $ Sepal Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Table: The structure of data set

|| || || ||

Data analysis

The histogram of each features for each Species are shown in Figure 1.

```
iris %>% gather(attributes, value, 1:4) %>%
  ggplot(aes(value, fill = attributes)) +
  geom_histogram(bins = 20,colour="black",alpha = 0.5) +
  facet_wrap(. ~ Species) +
  theme_light() +
  theme(legend.title = element_blank())
```

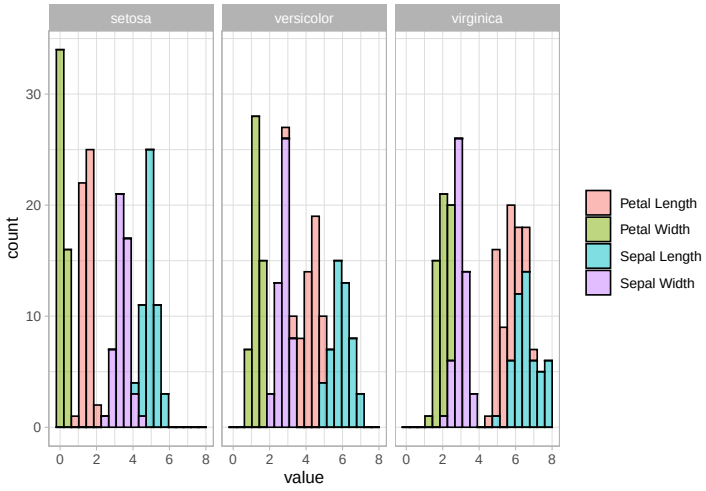

Figure 1: Histogram of features for each species

The Density of each Species at each of features also is shown in Figure 2.

```
iris %>% gather(attributes, value, 1:4) %>%
  ggplot(aes(value, fill = Species)) +
  geom_density(alpha = 0.5)+
  facet_wrap(. ~ attributes) +
  theme(legend.title =  element_blank()) +
  ylab("Density")+
  theme_light()
```

The boxplot of each features is shown in Figure 3.

```
iris %>% gather(attributes , value, 1:4) %>%
  ggplot(aes(attributes , value, fill = attributes)) +
  geom_boxplot() +
  theme_light() +
  theme(legend.title = element_blank(),
        axis.title.x = element_blank(),
        legend.position = "bottom")
```

The Correlation of each features in each Species is shown in Figure 4. The correlation of Sepal length with Petal length and width are high. Also in this figure the correlation in each Species based on each features are shown.

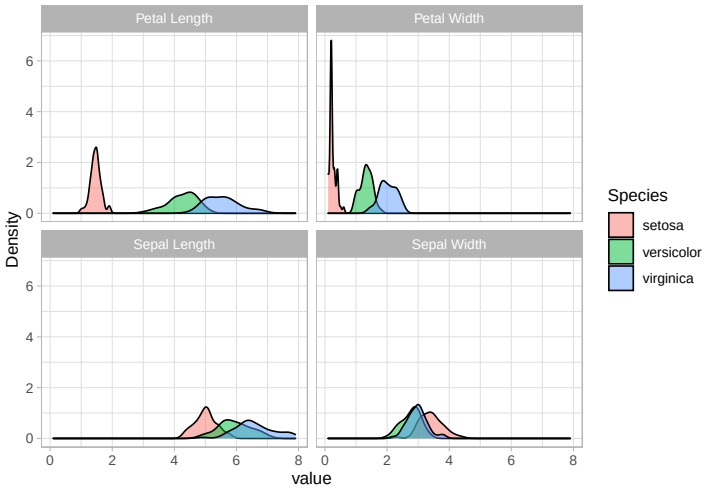

Figure 2: Density of each species in each features

Figure 3: Boxplot of each features

```
ggpairs(cbind(iris, Cluster=as.factor(iris$Species)),
        columns=1:4, aes(colour=Cluster, alpha=0.5),
        lower=list(continuous="points"),
        axisLabels="none", switch="both") +
  theme_light()
```

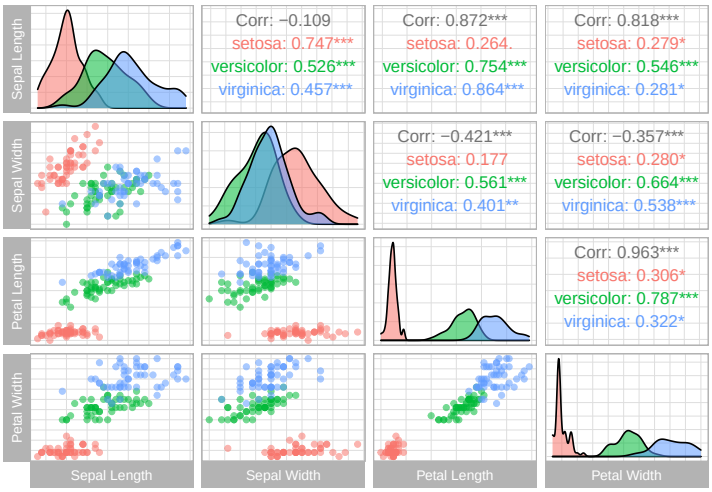

Figure 4: Correlation of each features in each Species

Data Prepration

Before we started to do the machine learning, it is betther some modification is done on data.

Data Normalized

In order to compare of each feature, it is better all features normalized.

```
iris_scaled <- scale(iris[,1:4])
final_iris <- cbind(iris_scaled,iris[,5])

kable(head(final_iris),
      "pandoc",
      caption = "The first six rows of normalized data",
      align = "c",
      font_size = 5)
```

Table 4: The first six rows of normalized data

Sepal Length	Sepal Width	Petal Length	Petal Width	Species
-0.8976739	1.0286113	-1.336794	-1.308593	setosa
-1.1392005	-0.1245404	-1.336794	-1.308593	setosa
-1.3807271	0.3367203	-1.393470	-1.308593	setosa
-1.5014904	0.1060900	-1.280118	-1.308593	setosa
-1.0184372	1.2592416	-1.336794	-1.308593	setosa

Sepal Length	Sepal Width	Petal Length	Petal Width	Species
-0.5353840	1.9511326	-1.166767	-1.046525	setosa

Spliting dataset to train and test

The data set is splitting to test and train with proporation of 30 to 70 percent.

```
set.seed(123)
test_index <- createDataPartition(final_iris$Species,times = 1, p= 0.3, list = FALSE)
train <- final_iris[-test_index,]
test <- final_iris[test_index,]
kable(cbind(trainDimention = dim(train), testDimention = dim(test)),
      "pandoc",
      caption = "The dimention of train and test data set",
      align = "c",
      font_size = 5)
```

Table 5: The dimention of train and test data set

trainDimention	testDimention
105	45
5	5

Training the KNN Algorithm

The K nearest neighbor (KNN) algorithm used to training the train data set. In this algorithm the best K value has to find on training data set.

```
fit_knn <- NULL
Accuracy <- NULL

for (i in 1:20){
  fit_knn <- knn(train[,1:4],test[,1:4], train$Species, k =i)
  Accuracy[i] <- mean(fit_knn == test$Species)
}
best_k <- which.max(Accuracy)
best_k
```

```
## [1] 12
```

Figure 5 shows the variation of Accuracy versus K value. The best K value is 6 with high Accuracy. The variation in Accuracy vesuse K is related to small size of dataset. Also the variation of Accuracy for K from 1 to 20, is around 5%.

```
k <- 1:20
Accuracy.df <- data.frame(Accuracy,k)

ggplot(Accuracy.df, aes(k, Accuracy)) + geom_point() +
     geom_line(lty = "dotted", color  = "red")
```

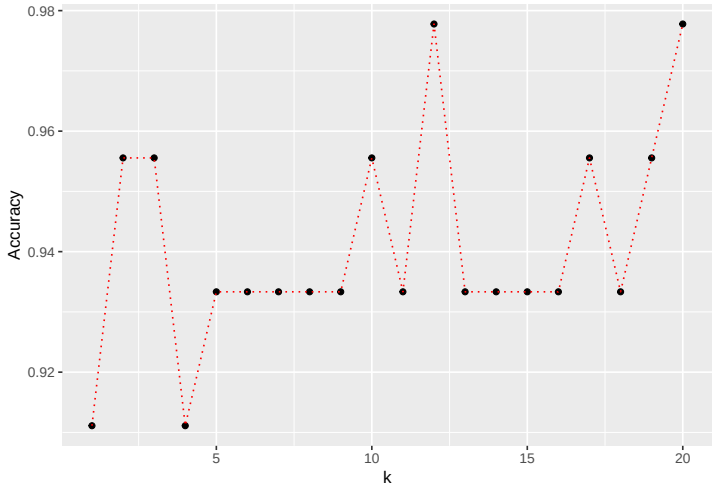

Figure 5: Variation of Accuracy versus K value

Evaluating the KNN Algorithm for Test

The test dataset is validated by KNN algorithm based on best K value around 6. The Accuracy of this algorithm is around 97.8%.

```
Accuracy[best_k]
```

```
## [1] 0.9777778
```