Supervised learning: Classification 2

Contents

Introduction 1

Confusion matrix, continued 1

Iris dataset 3

Classification trees 6

Final assignment: Random forest for classification 7

Introduction

In this practical, we will dive deeper into assessing classification methods and we will perform classification using tree-based methods.

We will use the packages `pROC`, `rpart`, `rpart.plot`, and `randomForest`. For this, you will probably need to `install.packages()` before running the `library()` functions.

```
library(MASS)
library(ISLR)
library(tidyverse)

library(pROC)

library(rpart)
library(rpart.plot)
library(randomForest)
```

Confusion matrix, continued

In the `data/` folder there is a cardiovascular disease dataset of 253 patients. The goal is to predict whether a patient will respond to treatment based on variables in this dataset:

- severity of the disease (low/high)
- age of the patient
- gender of the patient
- bad behaviour score (e.g. smoking/drinking)
- prior occurrence of the cardiovascular disease

- dose of the treatment administered: 1 (lowest), 2 (medium), or 3 (highest)

---

1. **Create a logistic regression model `lr_mod` for this data using the formula `response ~ .` and create a confusion matrix based on a .5 cutoff probability.**

Confusion matrix metrics

---

2. **Calculate the accuracy, true positive rate (sensitivity), the true negative rate (specificity), the false positive rate, the positive predictive value, and the negative predictive value. You can use the confusion matrix table on wikipedia. What can you say about the model performance? Which metrics are most relevant if this model were to be used in the real world?**

---

3. **Create an LDA model `lda_mod` for the same prediction problem. Compare its performance to the LR model.**

---

4. **Compare the classification performance of `lr_mod` and `lda_mod` for the new patients in the `data/new_patients.csv`.**

Brier score

Calculate the out-of-sample brier score for the `lr_mod` and give an interpretation of this number.

ROC curve

---

5. **Create two LR models: `lr1_mod` with `severity`, `age`, and `bb_score` as predictors, and `lr2_mod` with the formula `response ~ age + I(age^2) + gender + bb_score * prior_cvd * dose`. Save the predicted probabilities on the training data.**

---

6. **Use the function `roc()` from the `pROC` package to create two ROC objects with the predicted probabilities: `roc_lr1` and `roc_lr2`. Use the `ggroc()` method on these objects to create an ROC curve plot for each. Which model performs better? Why?**

7. **Print the `roc_lr1` and `roc_lr2` objects. Which AUC value is higher? How does this relate to the plots you made before? What is the minimum AUC value and what would a "perfect" AUC value be and how would it look in a plot?**

Iris dataset

One of the most famous classification datasets is a dataset used in R.A. Fisher's 1936 paper on linear discriminant analysis: the `iris` dataset. Fisher's goal was to classify the three subspecies of iris according to the attributes of the plants: `Sepal.Length`, `Sepal.Width`, `Petal.Length`, and `Petal.Width`:

The paper includes a hand-drawn graph worth looking at:

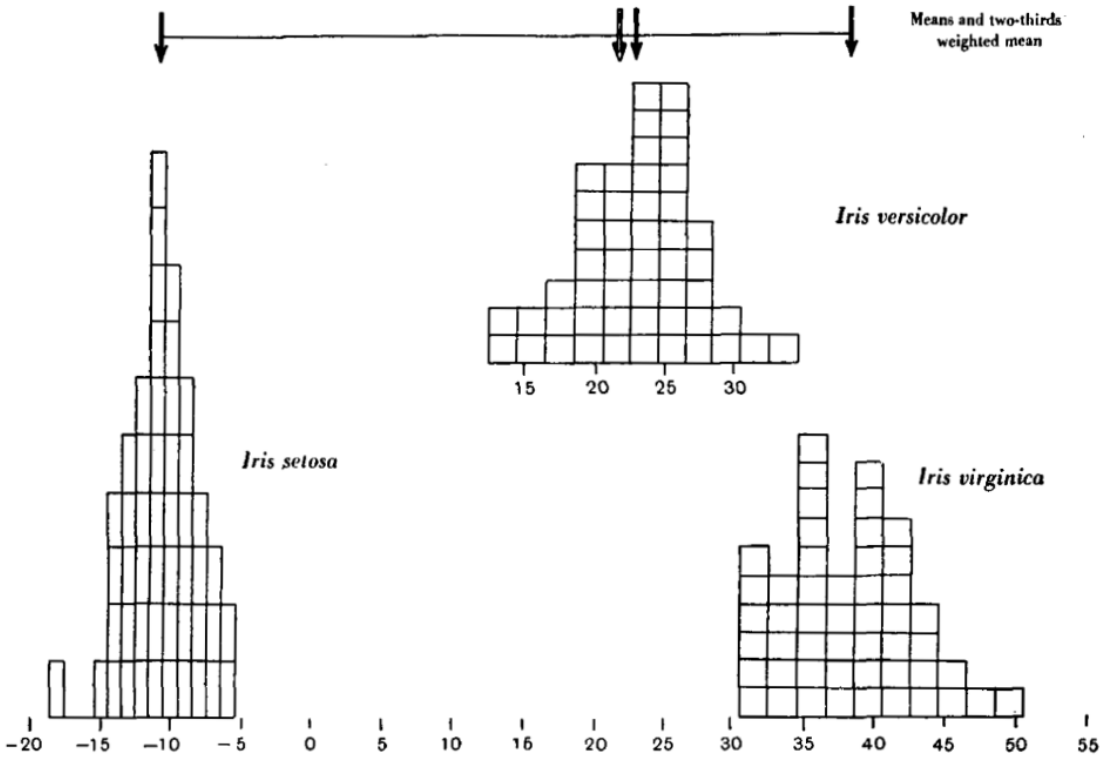


Fig. 1. Frequency histograms of the discriminating linear function, for three species of *Iris*.

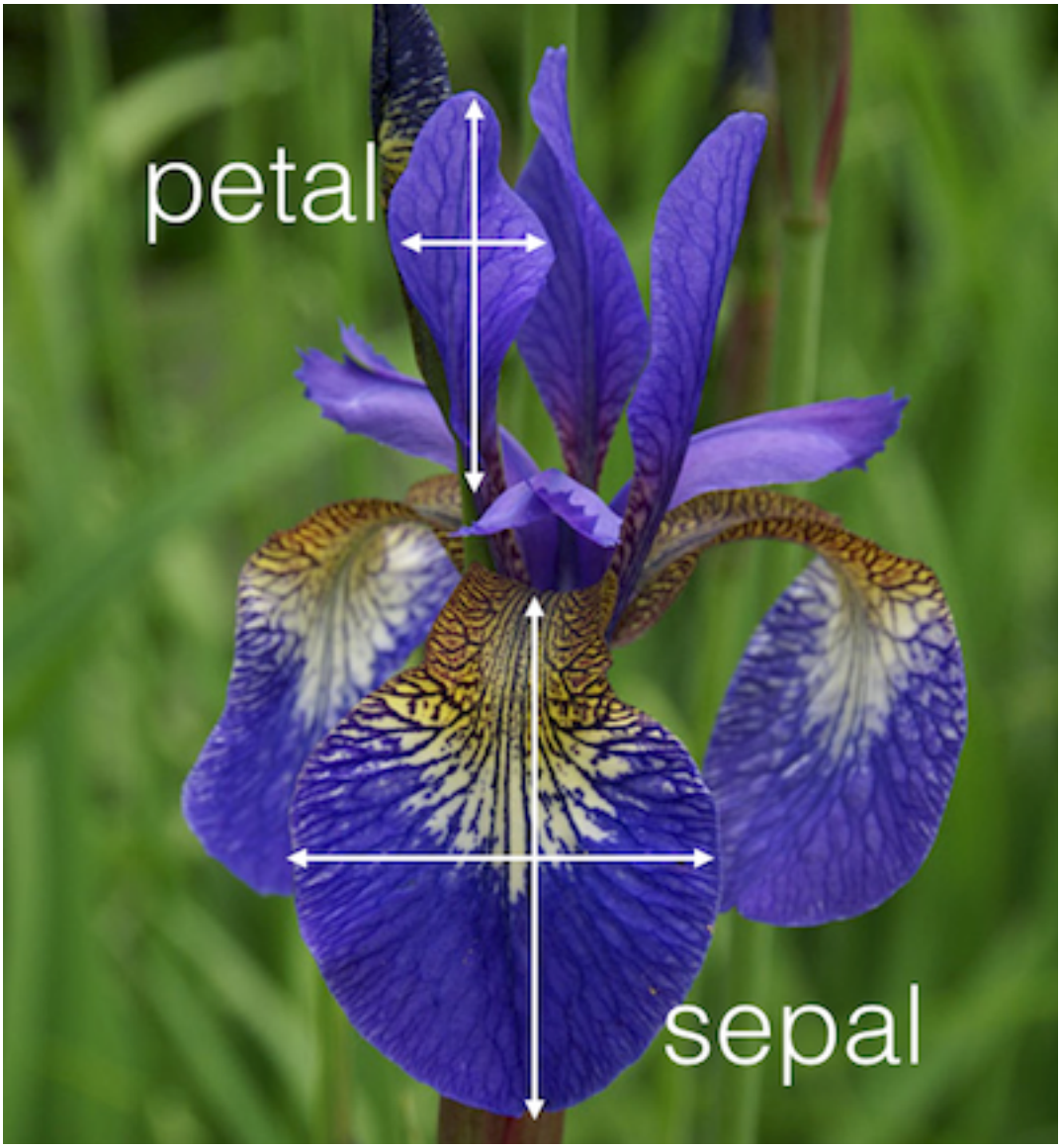


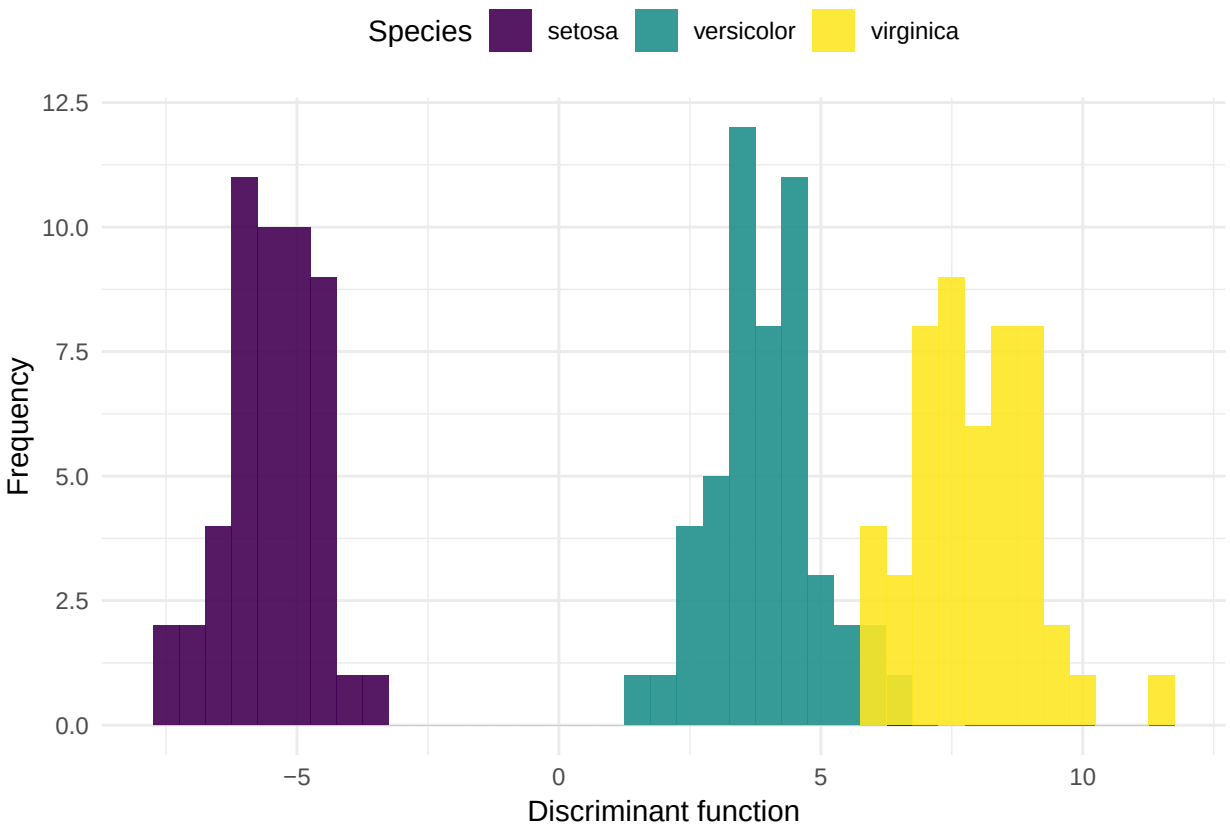
Figure 1: source: kaggle

We can reproduce this graph using the first linear discriminant from the `lda()` function:

```
# fit lda model, i.e. calculate model parameters
lda_iris <- lda(Species ~ ., data = iris)

# use those parameters to compute the first linear discriminant
first_ld <- -c(as.matrix(iris[, -5]) %*% lda_iris$scaling[,1])

# plot
tibble(
  ld = first_ld,
  Species = iris$Species
) %>%
  ggplot(aes(x = ld, fill = Species)) +
  geom_histogram(binwidth = .5, position = "identity", alpha = .9) +
  scale_fill_viridis_d(guide = ) +
  theme_minimal() +
  labs(
    x = "Discriminant function",
    y = "Frequency",
    main = "Fisher's linear discriminant function on Iris species"
  ) +
  theme(legend.position = "top")
```



8. **Explore the iris dataset using summaries and plots.**

9. **Fit an additional LDA model, but this time with only `Sepal.Length` and `Sepal.Width` as predictors. Call this model `lda_iris_sepal`**

10. **Create a confusion matrix of the `lda_iris` and `lda_iris_sepal` models. (NB: we did not split the dataset into training and test set, so use the training dataset to generate the predictions.). Which performs better in terms of accuracy?**

Classification trees

Classification trees in R can be fit using the `rpart()` function.

11. **Use `rpart()` to create a classification tree for the `Species` of `iris`. Call this model `iris_tree_mod`. Plot this model using `rpart.plot()`.**

12. **How would an iris with 2.7 cm long and 1.5 cm wide petals be classified?**

Because the classification tree only uses two variables, we can create another insightful plot using the splits on these variables.

13. **Create a scatterplot where you map `Petal.Length` to the x position and `Petal.Width` to the y position. Then, manually add a vertical and a horizontal line (using `geom_segment`) at the locations of the splits from the classification tree. Interpret this plot.**

There are several control parameters (tuning parameters) to the `rpart()` algorithm. You can find the available control parameters using `?rpart.control`.

14. **Create a classification tree model where the splits continue until all the observations have been classified. Call this model `iris_tree_full_mod`. Plot this model using `rpart.plot()`. Do you expect this model to perform better or worse on new Irises?**

Final assignment: Random forest for classification

15. **Use the function `randomForest()` to create a random forest model on the iris dataset. Use the function `importance()` on this model and create a bar plot of variable importance. Does this agree with your expectations? How well does the random forest model perform compared to the `lda_iris` model?**