Supervised learning: Classification 1

Contents

Introduction

In this practical, we will learn about nonlinear extensions to regression using basis functions and how to create, visualise, and interpret them. Parts of it are adapted from the practicals in ISLR chapter 7.

One of the packages we are going to use is `class`. For this, you will probably need to `install.packages("class")` before running the `library()` functions.

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

Default dataset

The default dataset contains credit card loan data for 10 000 people. The goal is to classify credit card cases as `yes` or `no` based on whether they will default on their loan.

---

1. **Create a scatterplot of the `Default` dataset, where `balance` is mapped to the x position, `income` is mapped to the y position, and `default` is mapped to the colour. Can you see any interesting patterns already?**

---

2. **Add `facet_grid(cols = vars(student))` to the plot. What do you see?**

3. **Transform "student" into a dummy variable using `ifelse()` (0 = not a student, 1 = student). Then, randomly split the Default dataset into a training set `default_train` (80%) and a test set `default_test` (20%)**

K-Nearest Neighbours

Now that we have explored the dataset, we can start on the task of classification. We can imagine a credit card company wanting to predict whether a customer will default on the loan so they can take steps to prevent this from happening.

The first method we will be using is k-nearest neighbours (KNN). It classifies datapoints based on a majority vote of the k points closest to it. In R, the `class` package contains a `knn()` function to perform knn.

4. **Create class predictions for the test set using the `knn()` function. Use `student`, `balance`, and `income` (but no basis functions of those variables) in the `default_train` dataset. Set k to 5. Store the predictions in a variable called `knn_5_pred`.**

5. **Create two scatter plots with income and balance as in the first plot you made. One with the true class (`default`) mapped to the colour aesthetic, and one with the predicted class (`knn_5_pred`) mapped to the colour aesthetic. Hint: Add the predicted class `knn_5_pred` to the `default_test` dataset before starting your `ggplot()` call of the second plot. What do you see?**

6. **Repeat the same steps, but now with a `knn_2_pred` vector generated from a 2-nearest neighbours algorithm. Are there any differences?**

Confusion matrix

The confusion matrix is an insightful summary of the plots we have made and the correct and incorrect classifications therein. A confusion matrix can be made in R with the `table()` function by entering two `factors`:

```
table(true = default_test$default, predicted = knn_2_pred)
```

```
##       predicted
## true     No  Yes
##   No   1878   51
##   Yes    52   19
```

---

7. **What would this confusion matrix look like if the classification were perfect?**

---

8. **Make a confusion matrix for the 5-nn model and compare it to that of the 2-nn model. What do you conclude?**

We will go more into the assessment of confusion matrices in the next practical.

Logistic regression

KNN directly predicts the class of a new observation using a majority vote of the existing observations closest to it. In contrast to this, logistic regression predicts the `log-odds` of belonging to category 1. These log-odds can then be transformed to probabilities by performing an inverse logit transform:

$$p = \frac{1}{1 + e^{-\alpha}}$$

, where α indicates log-odds for being in class 1 and p is the probability.

Therefore, logistic regression is a `probabilistic` classifier as opposed to a `direct` classifier such as KNN: indirectly, it outputs a probability which can then be used in conjunction with a cutoff (usually 0.5) to classify new observations.

Logistic regression in R happens with the `glm()` function, which stands for generalized linear model. Here we have to indicate that the residuals are modeled not as a gaussian (normal distribution), but as a `binomial` distribution.

---

9. **Use `glm()` with argument `family = binomial` to fit a logistic regression model `lr_mod` to the `default_train` data.**

Now we have generated a model, we can use the `predict()` method to output the estimated probabilities for each point in the training dataset. By default `predict` outputs the log-odds, but we can transform it back using the inverse logit function of before or setting the argument `type = "response"` within the predict function.

---

10. **Visualise the predicted probabilities versus observed class for the training dataset in `lr_mod`. You can choose for yourself which type of visualisation you would like to make. Write down your interpretations along with your plot.**

Another advantage of logistic regression is that we get coefficients we can interpret.

---

11. **Look at the coefficients of the `lr_mod` model and interpret the coefficient for `balance`. What would the probability of default be for a person who is not a student, has an income of 40000, and a balance of 3000 dollars at the end of each month? Is this what you expect based on the plots we've made before?**

Visualising the effect of the balance variable

In two steps, we will visualise the effect `balance` has on the predicted default probability.

---

12. **Create a data frame called `balance_df` with 3 columns and 500 rows: `student` always 0, `balance` ranging from 0 to 3000, and `income` always the mean income in the `default_train` dataset.**

---

13. **Use this dataset as the `newdata` in a `predict()` call using `lr_mod` to output the predicted probabilities for different values of `balance`. Then create a plot with the `balance_df$balance` variable mapped to x and the predicted probabilities mapped to y. Is this in line with what you expect?**

---

14. **Create a confusion matrix just as the one for the KNN models by using a cutoff predicted probability of 0.5. Does logistic regression perform better?**

Linear discriminant analysis

The last method we will use is LDA, using the `lda()` function from the `MASS` package.

15. **Train an LDA classifier `lda_mod` on the training set.**

16. **Look at the `lda_mod` object. What can you conclude about the characteristics of the people who default on their loans?**

17. **Create a confusion matrix and compare it to the previous methods.**

Final assignment

18. **Create a model (using knn, logistic regression, or LDA) to predict whether a 14 year old boy from the 3rd class would have survived the Titanic disaster. You can find the data in the `data/` folder. Would the passenger have survived if they were a girl in 2nd class?**