Comprehension Check: Generative Models

In the following exercises, we are going to apply LDA and QDA to the **`tissue_gene_expression`** dataset. We will start with simple examples based on this dataset and then develop a realistic example.

Q1

1/1 point (graded)

Create a dataset of samples from just cerebellum and hippocampus, two parts of the brain, and a predictor matrix with 10 randomly selected columns using the following code:

```
library(dslabs)
library(caret)
data("tissue_gene_expression")

set.seed(1993) #set.seed(1993, sample.kind="Rounding") if using R 3.6 or later
ind <- which(tissue_gene_expression$y %in% c("cerebellum", "hippocampus"))
y <- droplevels(tissue_gene_expression$y[ind])
x <- tissue_gene_expression$x[ind, ]
x <- x[, sample(ncol(x), 10)]
```

Use the `train` function to estimate the accuracy of LDA. For this question, use the entire `tissue_gene_expression` dataset: do not split it into training and test sets (understand this can lead to overfitting).

What is the accuracy?

0.8721386 ✓ **Answer:** 0.871

0.8721386

Explanation

The following code can be used to estimate the accuracy of the LDA:

```
fit_lda <- train(x, y, method = "lda")
fit_lda$results["Accuracy"]
```

Generating Speech Output

You have used 1 of 10 attempts

Submit

ℹ Answers are displayed within the problem

Q2

1/1 point (graded)

In this case, LDA fits two 10-dimensional normal distributions. Look at the fitted model by looking at the `finalModel` component of the result of `train`. Notice there is a component called `means` that includes the estimated `means` of both distributions. Plot the mean vectors against each other and determine which predictors (genes) appear to be driving the algorithm.

Which TWO genes appear to be driving the algorithm?

☐ PLCB1

☑ RAB1B

☐ MSH4

☑ OAZ2

☐ SPI1

☐ SAPCD1

☐ HEMK1

✓

Explanation

The following code can be used to make the plot:

```
t(fit_lda$finalModel$means) %>% data.frame() %>%
        mutate(predictor_name = rownames(.)) %>%
        ggplot(aes(cerebellum, hippocampus, label = predictor_name)) +
        geom_point() +
        geom_text() +
        geom_abline()
```

Submit You have used 1 of 3 attempts

Generating Speech Output

ℹ Answers are displayed within the problem

Q3

1/1 point (graded)

Repeat the exercise in Q1 with QDA.

Create a dataset of samples from just cerebellum and hippocampus, two parts of the brain, and a predictor matrix with 10 randomly selected columns using the following code:

```
library(dslabs)
library(caret)
data("tissue_gene_expression")

set.seed(1993) #set.seed(1993, sample.kind="Rounding") if using R 3.6 or later
ind <- which(tissue_gene_expression$y %in% c("cerebellum", "hippocampus"))
y <- droplevels(tissue_gene_expression$y[ind])
x <- tissue_gene_expression$x[ind, ]
x <- x[, sample(ncol(x), 10)]
```

Use the `train` function to estimate the accuracy of QDA. For this question, use the entire `tissue_gene_expression` dataset: do not split it into training and test sets (understand this can lead to overfitting).

What is the accuracy?

0.8147954 ✓ **Answer:** 0.815

0.8147954

Explanation

The following code can be used to estimate the accuracy of QDA:

```
fit_qda <- train(x, y, method = "qda")
fit_qda$results["Accuracy"]
```

Submit You have used 1 of 10 attempts

ℹ Answers are displayed within the problem

Q4

Generating Speech Output

Which TWO genes drive the algorithm when using QDA instead of LDA?

☐ PLCB1

☑ RAB1B

☐ MSH4

☑ OAZ2

☐ SPI1

☐ SAPCD1

☐ HEMK1

✓

Explanation

```
t(fit_qda$finalModel$means) %>% data.frame() %>%
        mutate(predictor_name = rownames(.)) %>%
        ggplot(aes(cerebellum, hippocampus, label = predictor_name)) +
        geom_point() +
        geom_text() +
        geom_abline()
```

The following code can be used to make the plot to evaluate which genes are driving the algorithm:

Submit   You have used 1 of 3 attempts

Answers are displayed within the problem

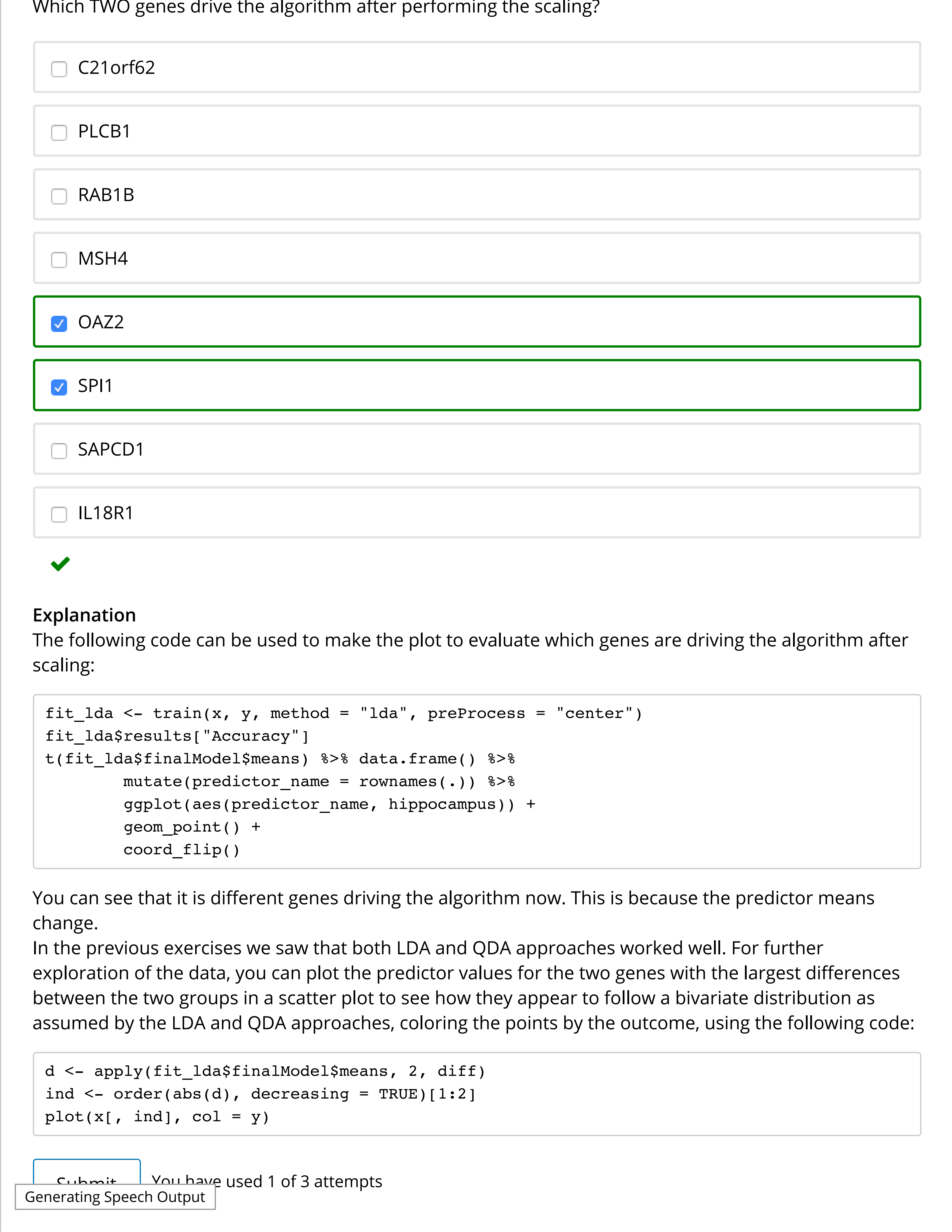
Q5

1/1 point (graded)

One thing we saw in the previous plots is that the values of the predictors correlate in both groups: some predictors are low in both groups and others high in both groups. The mean value of each predictor found in `colMeans(x)` is not informative or useful for prediction and often for purposes of interpretation, it is useful to center or scale each column. This can be achieved with the `preProcess` argument in `train`. Re-run LDA with `preProcess = "center"`. Note that accuracy does not change, but it is now easier to identify the predictors that differ more between groups than based on the plot made in Q2.

Generating Speech Output

Which TWO genes drive the algorithm after performing the scaling?



Which TWO genes drive the algorithm after performing the scaling?

☐ C21orf62

☐ PLCB1

☐ RAB1B

☐ MSH4

☑ OAZ2

☑ SPI1

☐ SAPCD1

☐ IL18R1

✓

Explanation

The following code can be used to make the plot to evaluate which genes are driving the algorithm after scaling:

```
fit_lda <- train(x, y, method = "lda", preProcess = "center")
fit_lda$results["Accuracy"]
t(fit_lda$finalModel$means) %>% data.frame() %>%
        mutate(predictor_name = rownames(.)) %>%
        ggplot(aes(predictor_name, hippocampus)) +
        geom_point() +
        coord_flip()
```

You can see that it is different genes driving the algorithm now. This is because the predictor means change.

In the previous exercises we saw that both LDA and QDA approaches worked well. For further exploration of the data, you can plot the predictor values for the two genes with the largest differences between the two groups in a scatter plot to see how they appear to follow a bivariate distribution as assumed by the LDA and QDA approaches, coloring the points by the outcome, using the following code:

```
d <- apply(fit_lda$finalModel$means, 2, diff)
ind <- order(abs(d), decreasing = TRUE)[1:2]
plot(x[, ind], col = y)
```

You have used 1 of 3 attempts

ℹ Answers are displayed within the problem

Q6

1/1 point (graded)

Now we are going to increase the complexity of the challenge slightly. Repeat the LDA analysis from Q5 but using all tissue types. Use the following code to create your dataset:

```
library(dslabs)
library(caret)
data("tissue_gene_expression")

set.seed(1993) #set.seed(1993, sample.kind="Rounding") if using R 3.6 or later
y <- tissue_gene_expression$y
x <- tissue_gene_expression$x
x <- x[, sample(ncol(x), 10)]
```

What is the accuracy using LDA?

0.8194837 ✓ **Answer:** 0.819

0.8194837

Explanation

The following code can be used to obtain the accuracy of the LDA:

```
fit_lda <- train(x, y, method = "lda", preProcess = c("center"))
fit_lda$results["Accuracy"]
```

We see that the results are slightly worse when looking at all of the tissue types instead of only selected ones. You can use the `confusionMatrix` function to learn more about what type of errors we are making, like this: `confusionMatrix(fit_lda)`.

Submit You have used 1 of 10 attempts

ℹ Answers are displayed within the problem

Ask your questions or make your comments about Generative Models here! **Remember, one of the best ways to reinforce your own learning is by explaining something to someone else, so we encourage you to answer each other's questions (without giving away the answers, of course).**

Some reminders:

Generating Speech Output

- Search the discussion board before posting to see if someone else has asked the same thing before asking a new question.
- Please be specific in the title and body of your post regarding which question you're asking about to facilitate answering your question.
- Posting snippets of code is okay, but posting full code solutions is not.
- If you do post snippets of code, please format it as code for readability. If you're not sure how to do this, there are instructions in a pinned post in the "general" discussion forum.

Discussion: Generative Models

Show Discussion

Topic: Section 4: Distance, Knn, Cross-Validation, and Generative Models / 4.3: Generative Models

Generating Speech Output