Assignment 2

Thomas Li

2025-03-27

Marking Rubric

On a high level:

- C grade (50%-65%): You have attempted the assignment, but in a limited way that might not show demonstrate full understanding of the concepts involved.
- B grade: (65%-79%) You have done what you were asked to do, and you did it correctly
- A grade (80%-100%): In addition to satisfying the requirements for a B grade, you have done something beyond, like:
    - exploring an additional avenue of research,
    - including work beyond the lecture content,
    - discussing current debates in the subject area,
    - demonstrating extraordinary investigative research or
    - writing an exceptional report,
    - …

You are usually not expected to write any long essays, so keep it brief, legible, and understandable. Support your arguments with figures, mathematical arguments, or references, where suitable.

In slightly more detail: Your assignment will be marked according to the following criteria. The relative weightings are only an approximate guide and will vary between questions.

Skills	Expectation for a C grade	(additional) Expectation for a B grade	(additional) Expectation for an A grade	approximate relative weighting
Communicating clearly via text	Text is in understandable English. For most questions, a few short sentences will be totally adequate, unless stated otherwise.	Uses adequate technical jargon, and concise language.	“Generic garbage” as sometimes produced by GenAI will lead to detractions here.	**necessary prerequisite.** Anything else cannot be marked if it is not communicated clear enough

Skills	Expectation for a C grade	(additional) Expectation for a B grade	(additional) Expectation for an A grade	approximate relative weighting
R Markdown and format of submission	Submission is a single file in either .html or .pdf format. File is generated by R Markdown. All code is shown in the html document.	—	—	**necessary prerequisite.**
R programming skills	Syntax correct, code runs without errors.	Code does what it is intended to do. Code is readable and uses code comments and suitable naming conventions to clarify what is being done. `dplyr` , `ggplot` , and statistical learning modules are employed in a suitable way.	Computations are reasonably efficient.	25%
Knowing and applying statistical learning algorithms	Basic statistical learning task is addressed.	Correct choice (within constraints of admissible modules) and application of algorithm.	—	30%
Interpreting and evaluating algorithm results and output	Some interpretation of the findings is provided, even if not correct.	Algorithm results are (when required) put into context, correctly explained, and implications as relevant for the assignment are correctly identified.	—	25%
Communicating clearly via figures	An attempt at visualisation (if required) is made.	Axes are labelled, figure is suitably scaled, intention is clearly communicated. Interpretation of the figure is given in either an expressive caption or in the accompanying full text.	—	10%

Skills	Expectation for a C grade	(additional) Expectation for a B grade	(additional) Expectation for an A grade	approximate relative weighting
Extra Effort	—	A good report will use the models and statistical methods to the extent developed in class.	A very good assignment (e.g., A- or better) will extend to something not covered, and/or show reflection beyond the narrow research question. Sometimes, question parts marked with “extra effort” give an indication for possible avenues to extend your work.	10%

Question A: Applying Logistic Regression to predict mortality from blood glucose and blood pressure

In this question, you are allowed to use `glm`.

We consider a heart health dataset in `heart.csv`. Your task is to predict `DEATH` (i.e., the patient died within the timeframe studied) from data `GLUCOSE` (amount of glucose in blood at last measurement) and `SYSBP` (systolic blood pressure at last measurement). This is a classification task. If you want to read up on what all of the other features mean, consult this documentation: https://biolincc.nhlbi.nih.gov/media/teachingstudies/FHS_Teaching_Longitudinal_Data_Documentation_2021a.pdf (https://biolincc.nhlbi.nih.gov/media/teachingstudies/FHS_Teaching_Longitudinal_Data_Documentation_2021a.pdf)

a. Split the dataset into a training set (80% of entries) and a test set (20% of entries).

b. Visualise the relationship between `DEATH` , `GLUCOSE` and `SYSBP` (s a suitable way. Form an initial hypothesis of what to look for when doing the classification.

c. On the training set, fit a (multiple) logistic regression model. Then:

   i. Compute the misclassification rates on the test set

   ii. Compute the confusion matrix on the test set

   iii. Visualise your fitted classification models suitable, e.g., by plotting the decision boundaries in the `GLUCOSE` - `SYSBP` -plane. Make a comment or observation regarding goodness of fit.

d. *Opportunities for showing extra effort:*

   - For public health purposes it is more important to catch *positives*, i.e. potential mortality risks, even if they end up not eventuating. In other words, false negatives are more dangerous than false positives.

     In order to address this problem, we can change the threshold at which an patient is classified as being “risky”: Instead of setting the decision boundary at probability $p = 50\%$, we classify a customer as “risky” (i.e., we predict `DEATH`) if the risk of them dying is higher than 10%. Modify your logistic regression to do this, and repeat the tasks of question c).

   - Compare the performance of logistic regression and discriminant analysis on this classification problem.

- Identify strong risk factors from this dataset and communicate your results.

Question B: Predicting color name from RGB values, using discriminant analysis

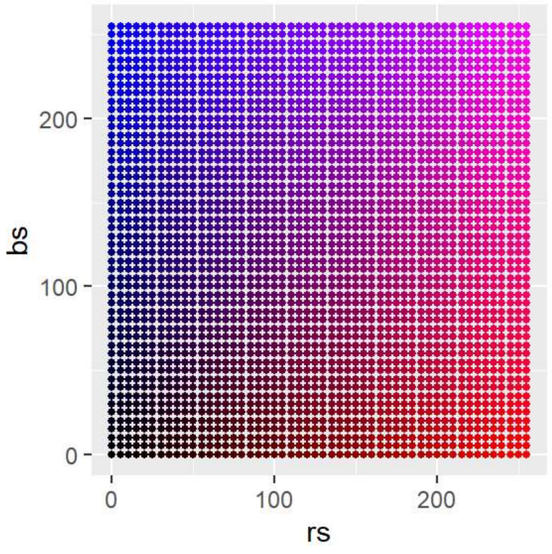
In this question, you are not allowed to use any pre-implemented modules for performing discriminant analysis, but you'll have to implement this yourself.

Colors can be coded via their RGB (red, green, blue) value in the form `(r,g,b)`, where `r`, `g`, and `b` are integers between 0 and 255. For example, `(255,0,0)` is pure red, and `(128,200,128)` is a shade of green.

In this exercise we will map `(r,g,b)` values to their color names. For example, we want `(255,0,0)` to be classified as red.

In order to make things a bit easier, we focus on the part of color space where `g=0`, i.e. there is no green component. This means the feature space is all combinations `(r,0,b)`, where `r` and `b` are between 0 and 255. For orientation, here is a visualisation of some of these colors, with each circle having the color of its `r` - `b` - coordinate.

```
rs <- seq(0,256,5)
bs <- seq(0,256,5)
df_plot_colors <- data.frame(rs = rs, bs=bs) %>% tidyr::expand(rs, bs)
ggplot(data=df_plot_colors) +
  geom_point(aes(x=rs, y=bs, color=rgb(rs/256,0,bs/256)), size=1)+# R's rgb code works with nu
mbers between 0 and 1 instead of between 0 and 255.
  scale_color_identity() +
  theme(legend.position = "none")
```



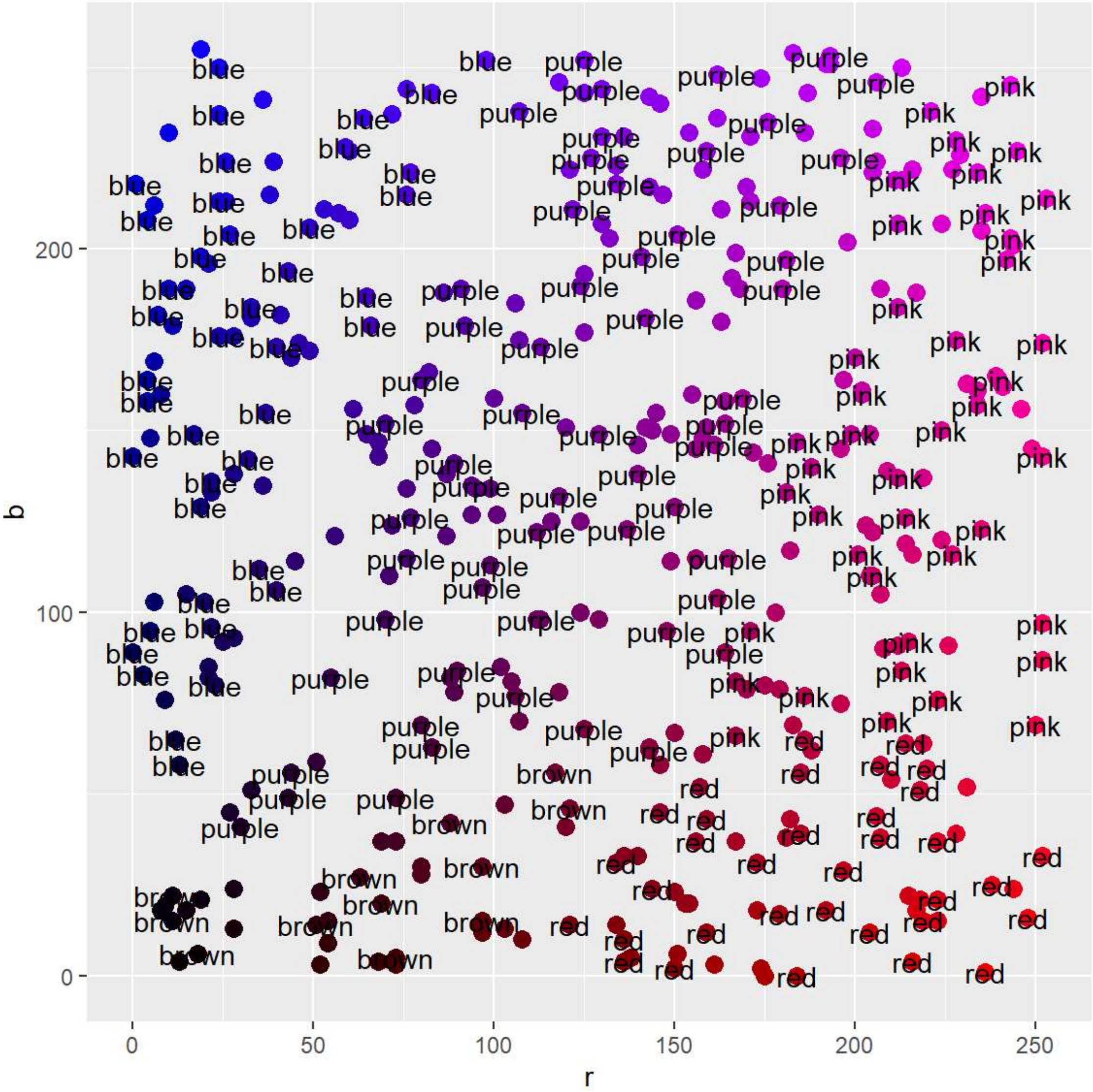
Our goal is to create a classification algorithm that takes an `r` value, and a `b` value, and outputs the name of the color this corresponds to.

Thankfully, we have a dataset where some of these labels have been entered. This is visualised below

```
df_colors <- read.csv("colors_train.csv")

df_colors_augmented <- df_colors %>% mutate(rgb = rgb(r/256,0,b/256))

ggplot(data=df_colors_augmented, aes(r, b, label = color)) +
  geom_point(aes(x=r, y=b, color=rgb), size=3) +
  geom_text(data = df_colors, check_overlap = TRUE) +
  scale_color_identity() +
  theme(legend.position = "none")
```



a. How many classes are there in the dataset?
b. Fit a QDA algorithm to this classification problem and visualise the decision boundaries in a suitable way.
c. Test your algorithm on `(200,0,200)`. What color is this being called by your algorithm?
d. *Opportunity for showing extra effort*: Learn about kNN-*Classification*, implement it, and compare it with your QDA algorithm's results.