Homework 5

Chance Kang / A13605546 / csk025@ucsd.edu
Jayden Kim / A16271107 / s0k003@ucsd.edu
Sia Sheth / A16357789 / snsheth@ucsd.edu
Math 189
Spring 2023

Conceptual Question

1. Linear Discriminant Analysis is a statistical tool used to classify data based on given factors. LDA uses classification to predict variables through theoretical deduction and assumptions. The iris flower dataset, an example of LDA, uses a combination of multiple variables to create a classifier into a species. We would use LDA when separating and determining classes, especially when the factors are independent, so that one factor does not directly affect another, skewing the classification. Logistic regression creates a binary classification model, modeling a relationship between covariates and the possibility of an event occurring or not. Logistic regression calculates the probability of the occurrence of an event based upon the independent variables. We would use logistic regression when the dependent variable has only two possible values (yes/no, success/failure, true/false, etc.) and would want to predict the probability of whether an event occurs or not.

Application Question

```
library(mlbench)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(tidyr)
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

2.

```
data(PimaIndiansDiabetes)
pima<-PimaIndiansDiabetes
```

a.

```
# 1 for pos; 0 for neg
pima$betenum <- 0
pima$betenum[pima$diabetes==unique(pima$diabetes)[1]] <- 1

# 1 for over 35; 0 for otherwise
pima$bmi35 <- 0
pima$bmi35 <- as.integer(pima$mass>35)

#fitting
diareg <- glm(betenum ~ glucose + bmi35, data = pima, family = "binomial")
summary(diareg)
```

```
##
## Call:
## glm(formula = betenum ~ glucose + bmi35, family = "binomial",
##     data = pima)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2489  -0.7792  -0.5215   0.8596   3.1177
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.44651    0.42633 -12.775  < 2e-16 ***
## glucose      0.03705    0.00327  11.330  < 2e-16 ***
## bmi35        0.59434    0.18242   3.258  0.00112 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 993.48  on 767  degrees of freedom
## Residual deviance: 798.17  on 765  degrees of freedom
## AIC: 804.17
##
## Number of Fisher Scoring iterations: 4
```

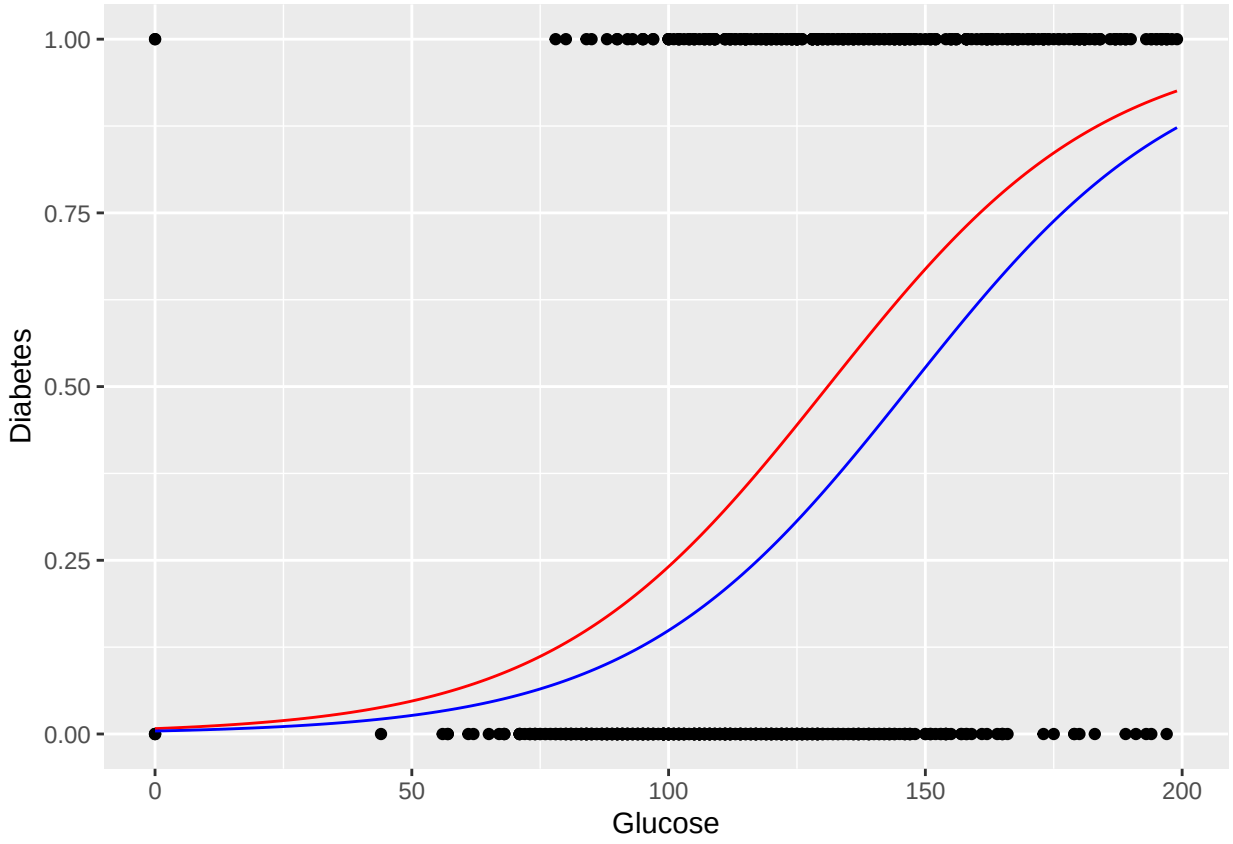
b.

```
xo <- cbind(rep(1,10000),seq(from = min(pima$glucose),to = max(pima$glucose),length.out = 1000), rep(1,
xu <- cbind(rep(1,10000),seq(from = min(pima$glucose),to = max(pima$glucose),length.out = 1000), rep(0,
pred_xo<-xo%*%coef(diareg)
pred_xu<-xu%*%coef(diareg)
pred_o <- 1/(1+exp(-pred_xo))
pred_u <- 1/(1+exp(-pred_xu))

pred_o_df <- data.frame(x = xo[,2], pred_o = pred_o)
pred_u_df <- data.frame(x = xu[,2], pred_u = pred_u)

ggplot() +
```

```
  geom_point(data=pima,aes(y=betenum,x=glucose)) +
  geom_line(data=pred_o_df,aes(x=x,y=pred_o),color="Red") +
  geom_line(data=pred_u_df,aes(x=x,y=pred_u),color="Blue") +
  labs(y="Diabetes",x="Glucose")
```



c.

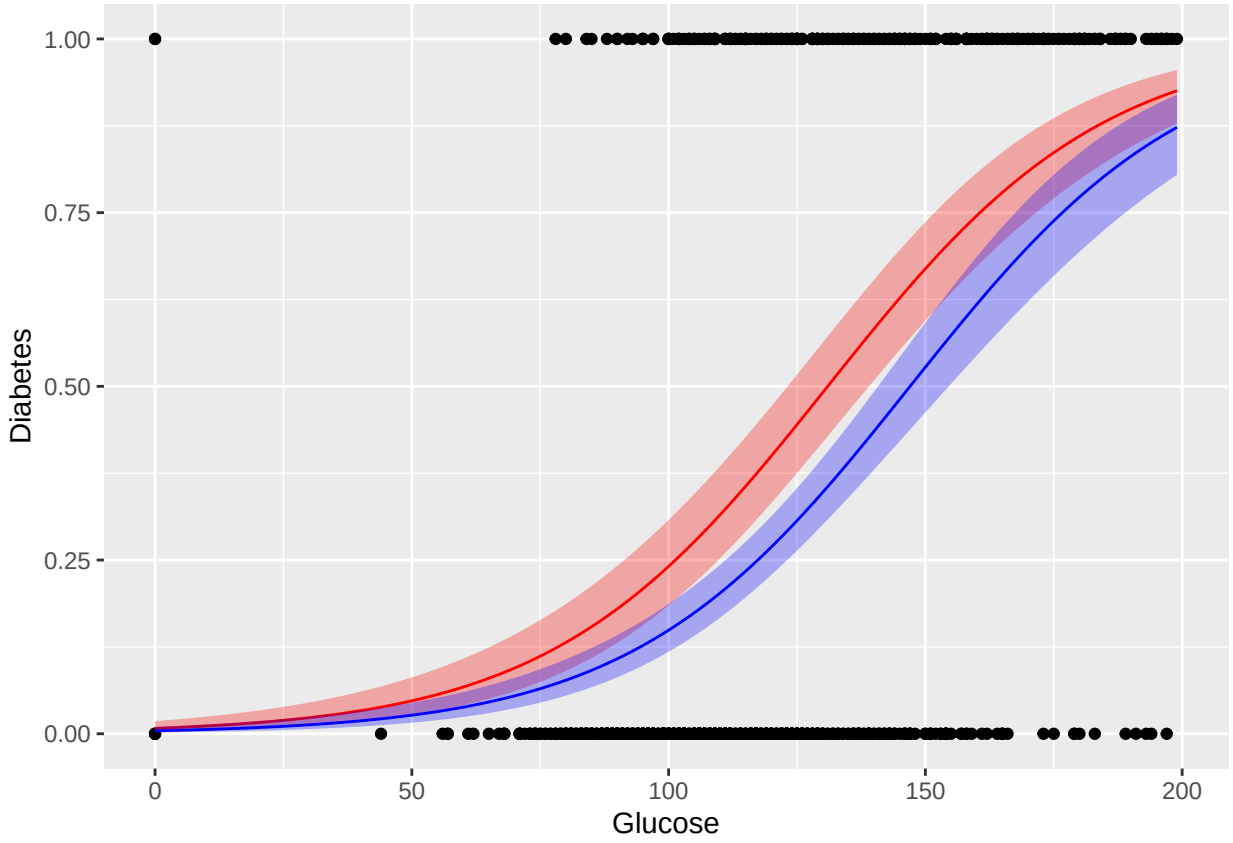
```
xo <- cbind(rep(1,10000),seq(from = min(pima$glucose),to = max(pima$glucose),length.out = 1000), rep(1,
xu <- cbind(rep(1,10000),seq(from = min(pima$glucose),to = max(pima$glucose),length.out = 1000), rep(0,
pred_xo<-xo%*%coef(diareg)
pred_xu<-xu%*%coef(diareg)
pred_o <- 1/(1+exp(-pred_xo))
pred_u <- 1/(1+exp(-pred_xu))

pred_o_df <- data.frame(x = xo[,2], pred_o = pred_o)
pred_u_df <- data.frame(x = xu[,2], pred_u = pred_u)

se_xo <- sqrt(diag(xo%*%vcov(diareg)%*%t(xo)))
se_xu <- sqrt(diag(xu%*%vcov(diareg)%*%t(xu)))
ci_xo <- cbind(pred_xo-1.96*se_xo,pred_xo+1.96*se_xo)
ci_xu <- cbind(pred_xu-1.96*se_xu,pred_xu+1.96*se_xu)
ci_o <- 1/(1+exp(-ci_xo))
ci_u <- 1/(1+exp(-ci_xu))
pred_o_df <- pred_o_df %>% mutate(lb=ci_o[,1],ub=ci_o[,2])
pred_u_df <- pred_u_df %>% mutate(lb=ci_u[,1],ub=ci_u[,2])

ggplot() +
```

```
  geom_point(data=pima,aes(y=betenum,x=glucose)) +
  geom_line(data=pred_o_df,aes(x=x,y=pred_o),color="Red") +
  geom_ribbon(data=pred_o_df,aes(x=x,ymin=lb,ymax=ub),alpha=0.3,fill="Red") +
  geom_line(data=pred_u_df,aes(x=x,y=pred_u),color="Blue") +
  geom_ribbon(data=pred_u_df,aes(x=x,ymin=lb,ymax=ub),alpha=0.3,fill="Blue") +
  labs(y="Diabetes",x="Glucose")
```



3.

a.

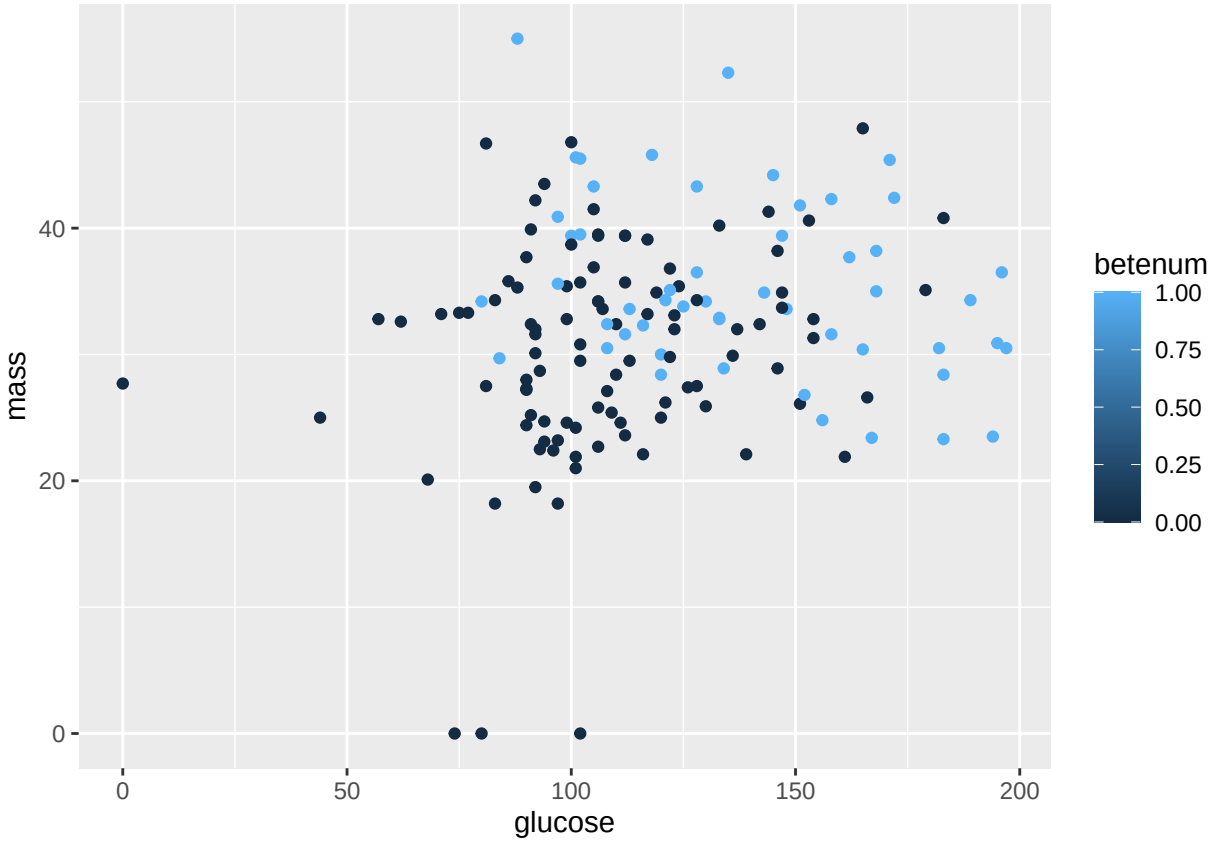
```
set.seed(123)
train <- sample(1:nrow(pima), nrow(pima)*0.8)
test <- (-train)
dialda <- lda(betenum ~ glucose + mass, data = pima, subset = train)
dialda
```

```
## Call:
## lda(betenum ~ glucose + mass, data = pima, subset = train)
##
## Prior probabilities of groups:
##         0         1
## 0.6482085 0.3517915
##
## Group means:
##    glucose     mass
## 0 110.4271 30.29749
## 1 141.8843 35.05093
```

```
##
## Coefficients of linear discriminants:
##                LD1
## glucose 0.03081278
## mass    0.05621735
```

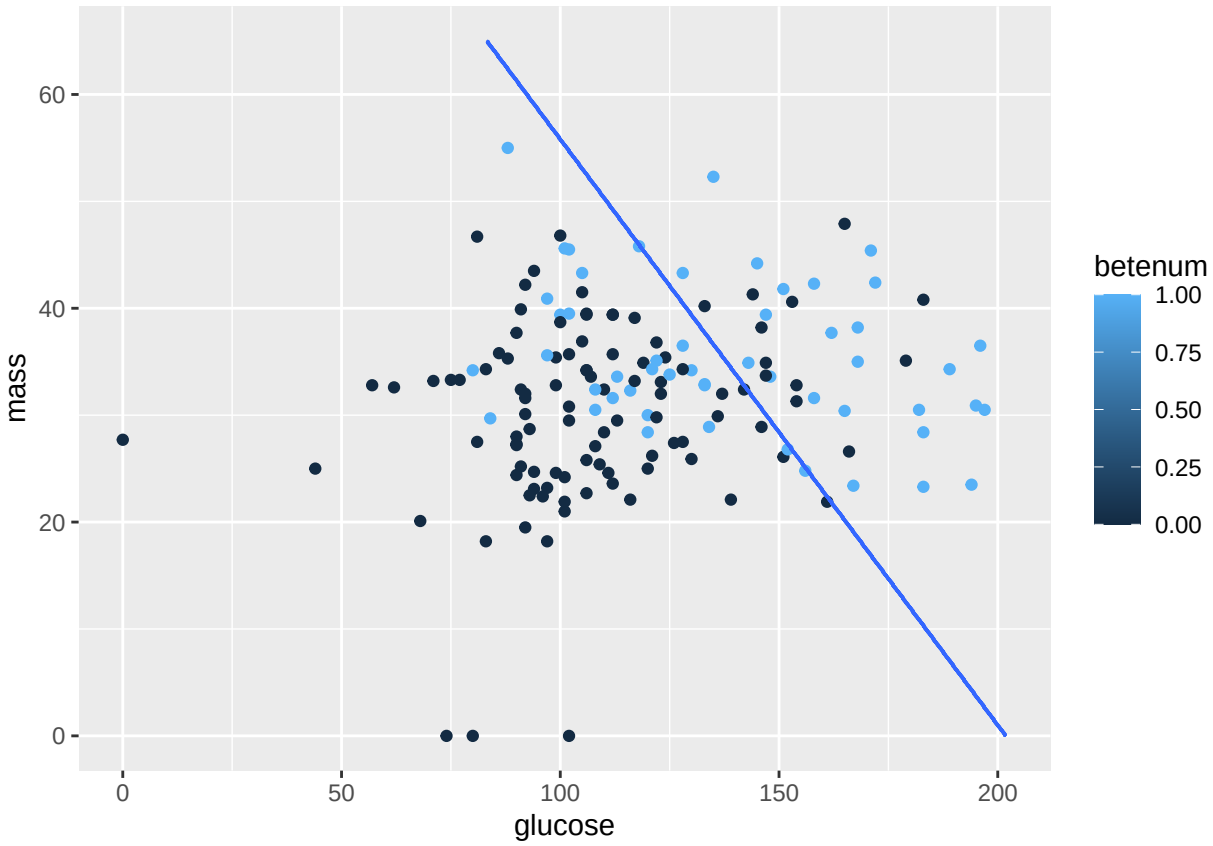
b.

```
ggplot() +
  geom_point(data = pima[test,], aes(x = glucose, y = mass, color = betenum))
```



c.

```
#x1 = glucose
grid <- expand.grid(x1 = seq(0, 250, length.out=1000), x2 = seq(0, 65, length.out=1000))
lad_pred <- predict(dialda, data.frame(glucose = grid$x1, mass = grid$x2))
grid$contours <- as.numeric(lad_pred$class == 1)
ggplot() +
  geom_point(data = pima[test,], aes(x = glucose, y = mass, color = betenum)) +
  geom_contour(data = grid, aes(x = x1, y = x2, z = contours))
```



d.

```
bete_act = pima$betenum[test]
lda_class <- predict(dialda, pima[test,])$class
table(lda_class, bete_act)
```

```
##          bete_act
## lda_class  0  1
##         0 90 28
##         1 12 24
```

```
mean(lda_class == bete_act)
```

```
## [1] 0.7402597
```

LDA model returned 28 false negatives and 12 false positives with its accuracy being around 72 percent. Though no false negatives would be ideal for a medical diagnose prediction, LDA has performed well with its relatively high accuracy.

4.

a.

```
diaqda<- qda(betenum ~ glucose + mass, data = pima, subset = train)
diaqda
```

```
## Call:
## qda(betenum ~ glucose + mass, data = pima, subset = train)
##
## Prior probabilities of groups:
##         0         1
```

```
## 0.6482085 0.3517915
##
## Group means:
##    glucose     mass
## 0 110.4271 30.29749
## 1 141.8843 35.05093
```

b.

```
qda_class <- predict(diaqda, pima[test,])$class
table(qda_class, bete_act)
```

```
##          bete_act
## qda_class  0  1
##         0 89 27
##         1 13 25
```

```
mean(qda_class == bete_act)
```

```
## [1] 0.7402597
```

c. Both models have the exact same accuracy of 74 percent. As for errors, QDA returned one more type 1 error than LDA whereas LDA returned one more type 2 error than QDA. Such means that LDA failed to identify one more individual with diabetes, incorrectly classifying them as not having it. As mentioned earlier, minimizing type 2 error would be ideal for a medical diagnose prediction and since the accuracy for both models are exactly the same, we would recommend utilizing QDA over LDA. Note that both model have performed well and similar.

Contribution

Our group homework process goes as the following:
1. Each member attempts to complete the homework
2. Compare and discuss the answers
3. Complete a finalized version to submit

All members have contributed about the equal amount to complete this homework.