Homework #2

Spencer Pease

2/08/2021

Question 1

Part (a)

Table 1: Observations by diagnosis class

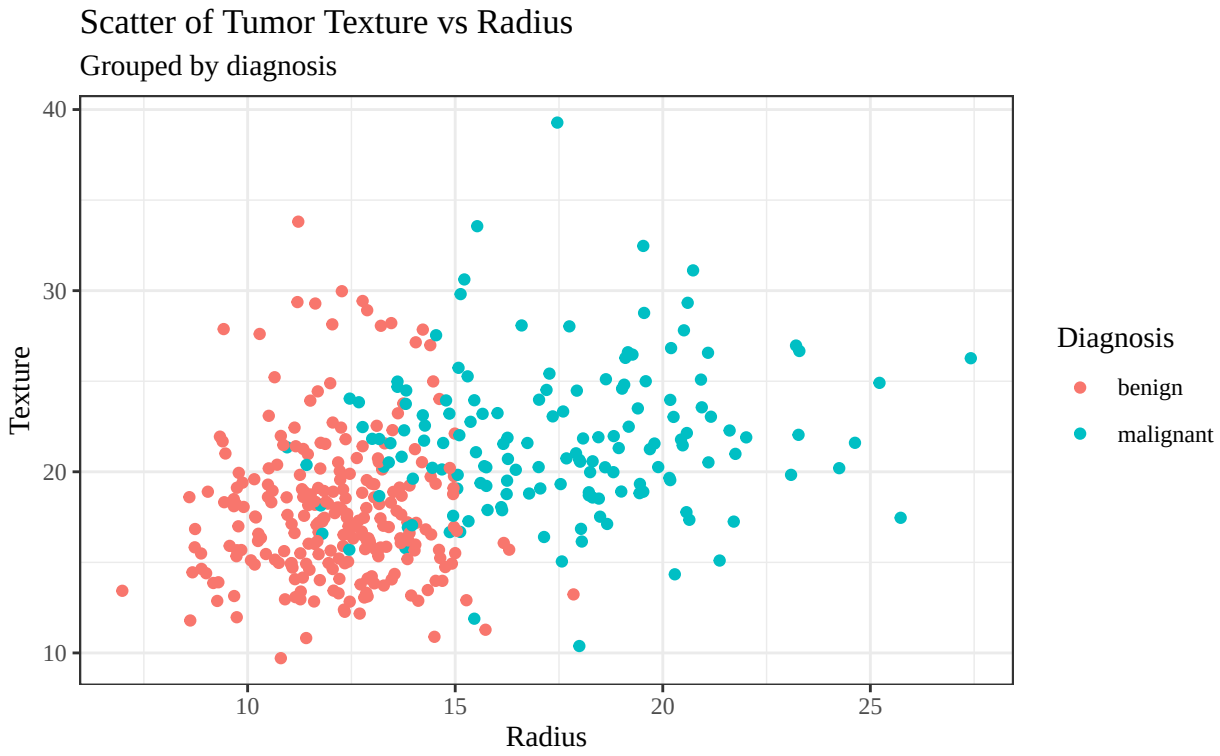
diagnosis	observations
benign	357
malignant	212

The *wdbc* dataset has a sample size n of 569 and 2 predictors p. *Table 1* shows a summary of the number of observations in each class.

Part (b)

This dataset is randomly split into a training set of 400 observations and a test set of 169 observations.

Part (c)



Based on the scatter of outcome and predictors, being able to predict with perfect accuracy looks impossible. The distributions of outcome classifications in each predictor dimension overlap in such a way as to eliminate any possible decision boundary that would be robust to new observations.

Part (d)

Fitting a logistic regression model of the form

$$\text{diagnosis} \sim \text{radius} + \text{texture}$$

to the training dataset produces results shown in the below table.

Table 2: Logistic regression summary

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-20.081	2.154	-9.324	0
radius	1.096	0.126	8.722	0
texture	0.205	0.043	4.725	0

From this model, a one unit increase in *radius* is associated with a 1.10 increase in the log-odds of the diagnosis being malignant, and a one unit increase in *texture* is associated with a 0.20 increase in the log-odds of a malignant diagnosis.

Part (e)

First, define p as the the predicted probability of a malignant diagnosis given an observed *radius* of 10 and *texture* of 12:

$$p = P(Y = M \mid (X_1, X_2) = (10, 12))$$

With these values and the coefficients of the logistic model defined above, the relationship between the input data and output probability is:

$$\text{logit}(p) = \beta_0 + \beta_1 X_1 + \beta_2 X_2$$

With this relation and the definition of the logit, the value of p is calculated to be:

$$\begin{aligned}
\frac{1-p}{p} &= \beta_0 + \beta_1 X_1 + \beta_2 X_2 \\
p &= \frac{\exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2)}{1 + \exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2)} \\
p &= \frac{\exp(-6.6603)}{1 + \exp(-6.6603)} \\
p &= 0.0013
\end{aligned}$$

This direct calculation agrees with the predicted probability computed directly with `predict()`, 0.0013.

Part (f)

If we know the estimated log-odds of the model to be 0.7, then the estimated probability of a malignant diagnosis can be calculated by applying the inverse logit function:

$$p = \text{invlogit}(0.7) = \frac{\exp(0.7)}{1 + \exp(0.7)} = 0.67$$

Part (g)

Table 3: Confusion matrix of predicted outcomes for the training dataset

obs / pred	benign	malignant
benign	233	15
malignant	29	123

Table 4: Confusion matrix of predicted outcomes for the test dataset

obs / pred	benign	malignant
benign	103	6
malignant	10	50

Performance metrics:

- Training accuracy: 0.89
- Test accuracy: 0.905

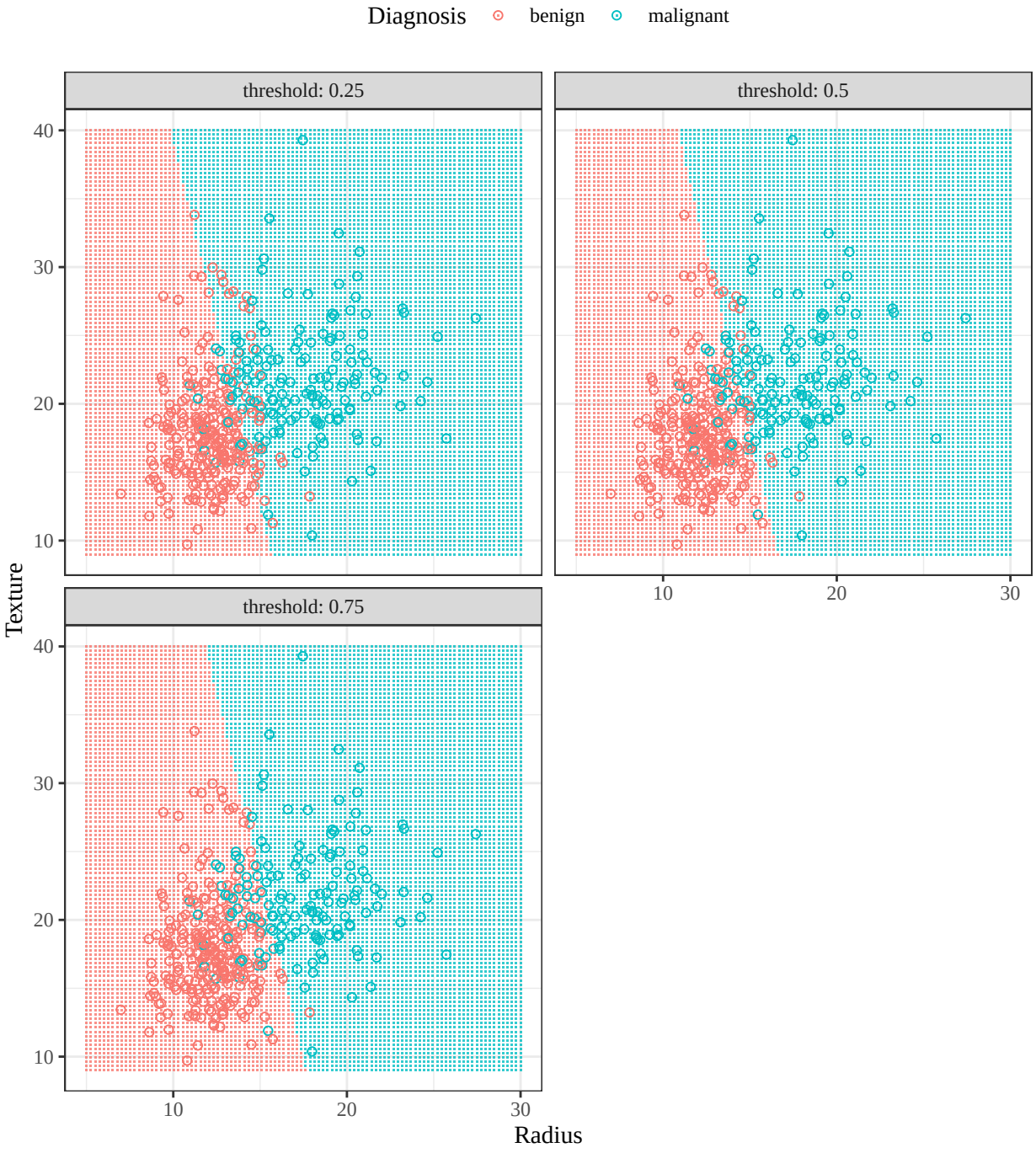
The prediction accuracy are similar for both the training and test datasets, and the confusion matrices both show this model is more likely to report false negatives than false positives on similar data distributions.

Part (h)

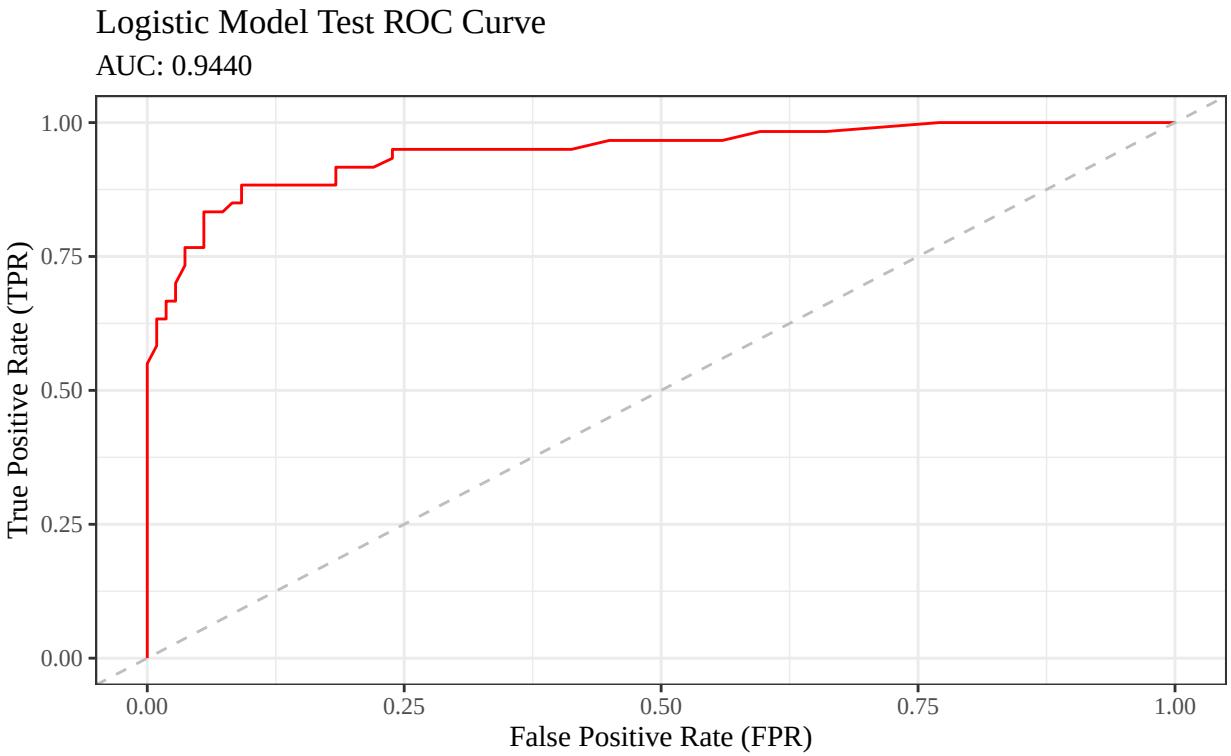
Plotting the decision boundary at different thresholds, we see the estimated boundary correctly classify more training observations with a *benign* diagnosis as the threshold increases, at the cost of mis-classifying more *malignant* diagnoses.

Decision Boundary of the logistic model

With different classification thresholds



Part (i)



Part (j)

The estimated area under the ROC curve computed on the test dataset for the logistic model is: 0.944.

Question 2

Part (a)

Table 5: LDA model prior probabilities and group means

diagnosis	prior	radius	texture
benign	0.62	12.127	17.852
malignant	0.38	17.476	21.678

For the linear discriminant analysis (LDA) model, the priors represent the probability of randomly selecting an observation of the given class from the data used to fit the model. The group means represent the center of the Gaussian distribution defining the likelihood for each diagnosis class. For LDA, it is assumed that these likelihoods have the same variance. Together, the priors and likelihoods are proportional to the posterior probabilities.

Part (b)

Table 6: Confusion matrix of predicted outcomes for the training dataset

obs / pred	benign	malignant
benign	239	9
malignant	37	115

Table 7: Confusion matrix of predicted outcomes for the test dataset

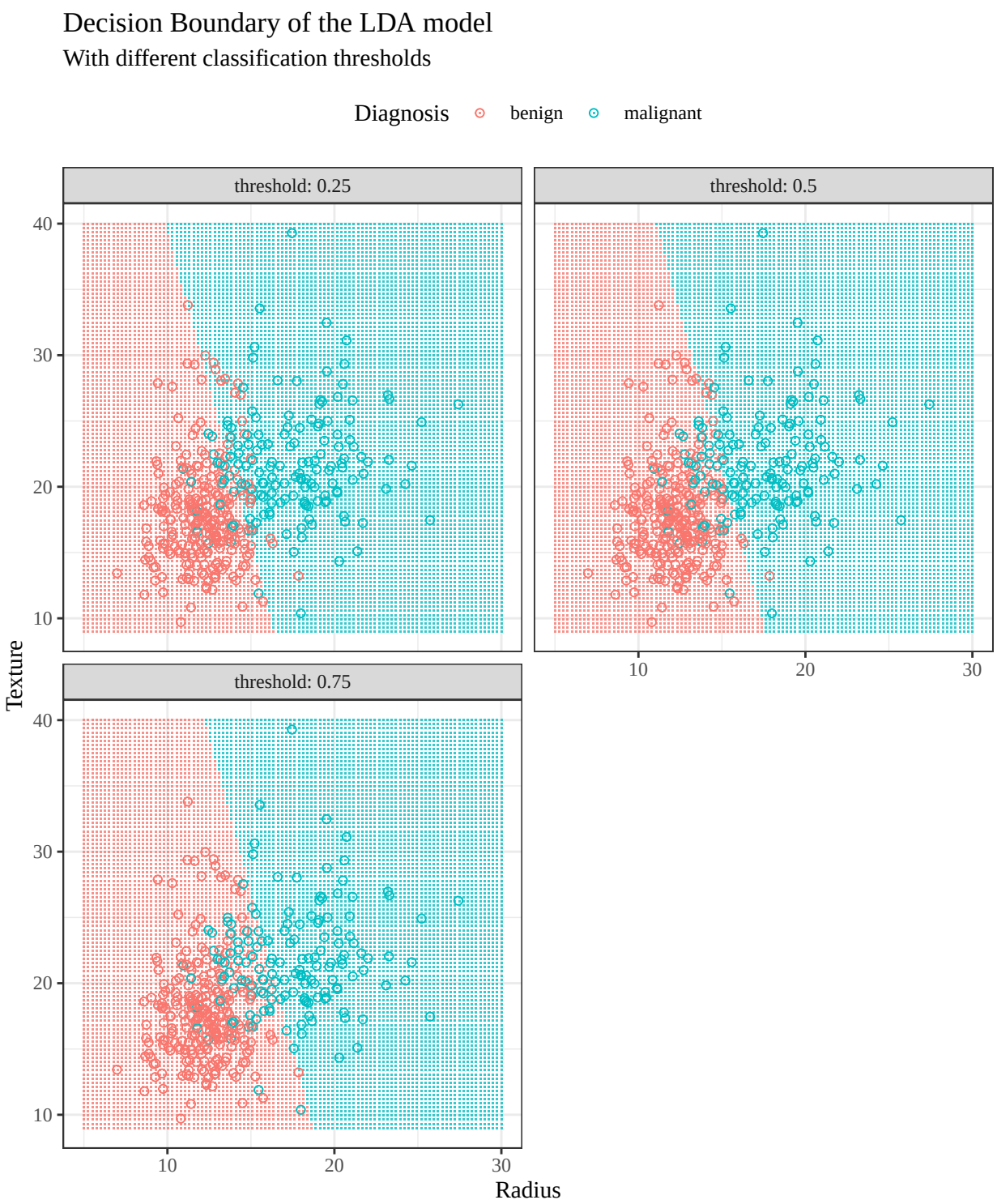
obs / pred	benign	malignant
benign	105	4
malignant	15	45

Performance metrics:

- Training accuracy: 0.885
- Test accuracy: 0.888

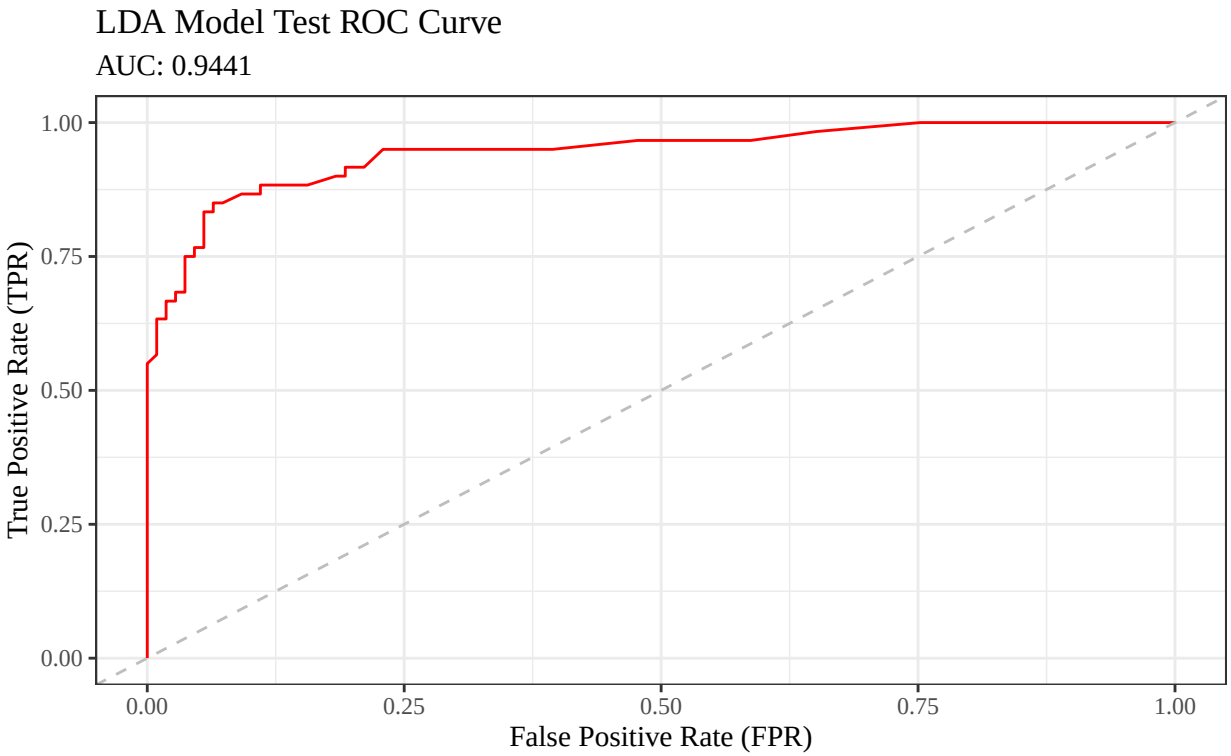
The predictive accuracy of the training and test set are similar to each other. This is reflected in the confusion matrices, which have similar proportions of correct and incorrect predictions.

Part (c)



Plotting the decision boundary at different thresholds, we see the estimated boundary correctly classify more training observations with a *benign* diagnosis as the threshold increases, at the cost of mis-classifying more *malignant* diagnoses.

Part (d)



Part (e)

The estimated area under the ROC curve computed on the test dataset for the logistic model is: 0.9441.

Question 3

Part (a)

Table 8: QDA model prior probabilities and group means

diagnosis	prior	radius	texture
benign	0.62	12.127	17.852
malignant	0.38	17.476	21.678

For the quadratic discriminant analysis (QDA) model, the priors represent the probability of randomly selecting an observation of the given class from the data used to fit the model. The group means represent the center of the Gaussian distribution defining the likelihood for each diagnosis class. For QDA, it is not assumed that these likelihoods have the same variance. Together, the priors and likelihoods are proportional to the posterior probabilities.

Part (b)

Table 9: Confusion matrix of predicted outcomes for the training dataset

obs / pred	benign	malignant
benign	238	10
malignant	35	117

Table 10: Confusion matrix of predicted outcomes for the test dataset

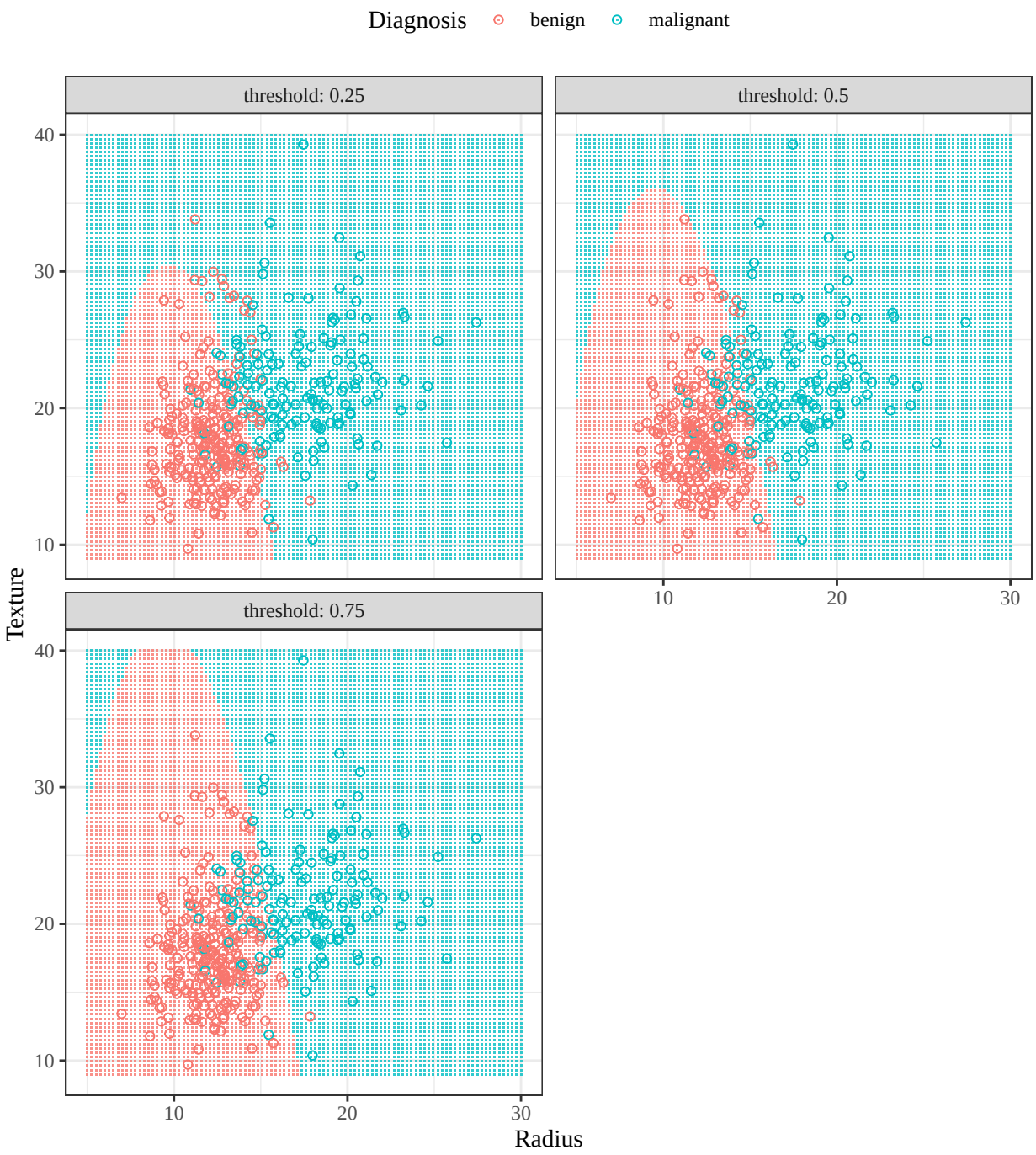
obs / pred	benign	malignant
benign	103	6
malignant	14	46

Performance metrics:

- Training accuracy: 0.887
- Test accuracy: 0.882

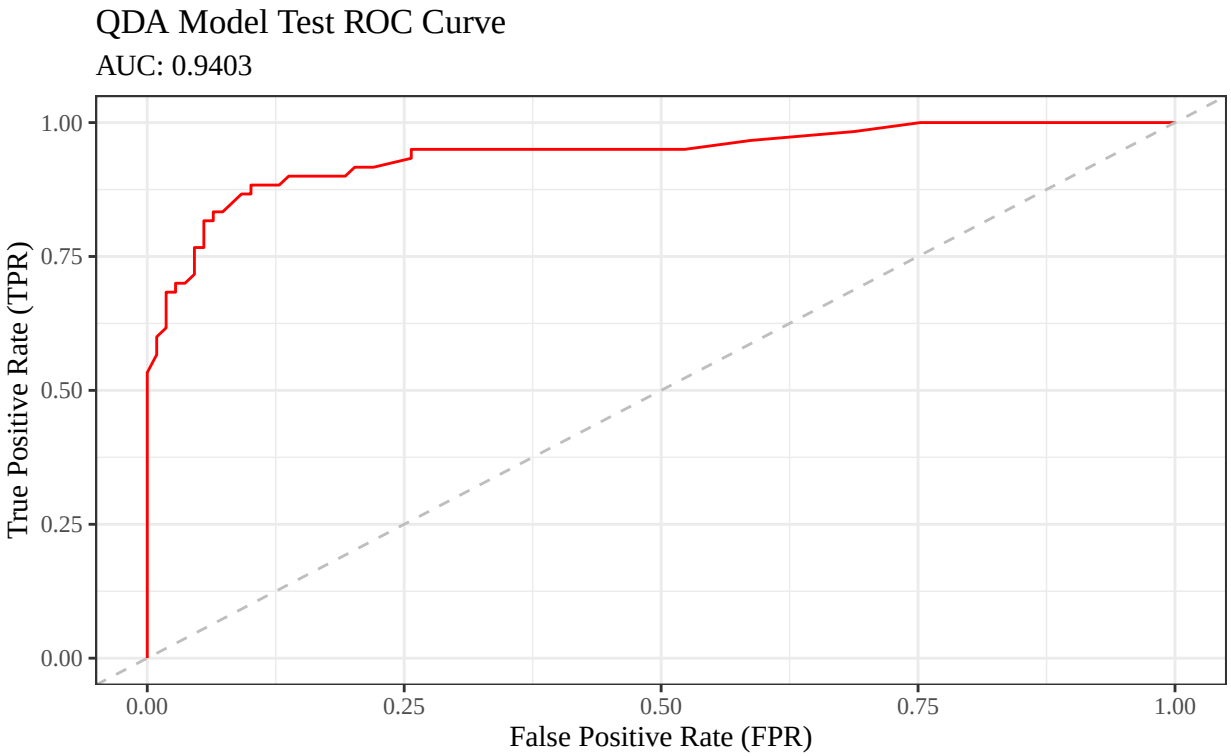
The overall performance between the training and test models is similar, with the predictive accuracies being almost identical.

Part (c)



As the threshold for a malignant diagnosis classification increases, the quadratic decision boundary shifts to mis-classify more malignant observations. The spread of data doesn't lend itself well to a quadratic decision boundary, and ends up practically functioning as a linear decision boundary.

Part (d)



Part (e)

The estimated area under the ROC curve computed on the test dataset for the logistic model is: 0.9403.

Question 4

Part (a)

Table 11: Training kNN confusion matrix (k = 1)

obs / pred	benign	malignant
benign	248	0
malignant	0	152

Table 12: Training kNN confusion matrix (k = 2)

obs / pred	benign	malignant
benign	233	15
malignant	16	136

Table 13: Training kNN confusion matrix (k = 3)

obs / pred	benign	malignant
benign	239	9
malignant	16	136

Table 14: Training kNN confusion matrix (k = 4)

obs / pred	benign	malignant
benign	234	14
malignant	20	132

Table 15: Training kNN confusion matrix (k = 20)

obs / pred	benign	malignant
benign	233	15
malignant	26	126

Table 16: Test kNN confusion matrix (k = 1)

obs / pred	benign	malignant
benign	93	16
malignant	10	50

Table 17: Test kNN confusion matrix (k = 2)

obs / pred	benign	malignant
benign	91	18
malignant	10	50

Table 18: Test kNN confusion matrix (k = 3)

obs / pred	benign	malignant
benign	98	11
malignant	8	52

Table 19: Test kNN confusion matrix (k = 4)

obs / pred	benign	malignant
benign	99	10
malignant	10	50

Table 20: Test kNN confusion matrix (k = 20)

obs / pred	benign	malignant
benign	101	8
malignant	9	51

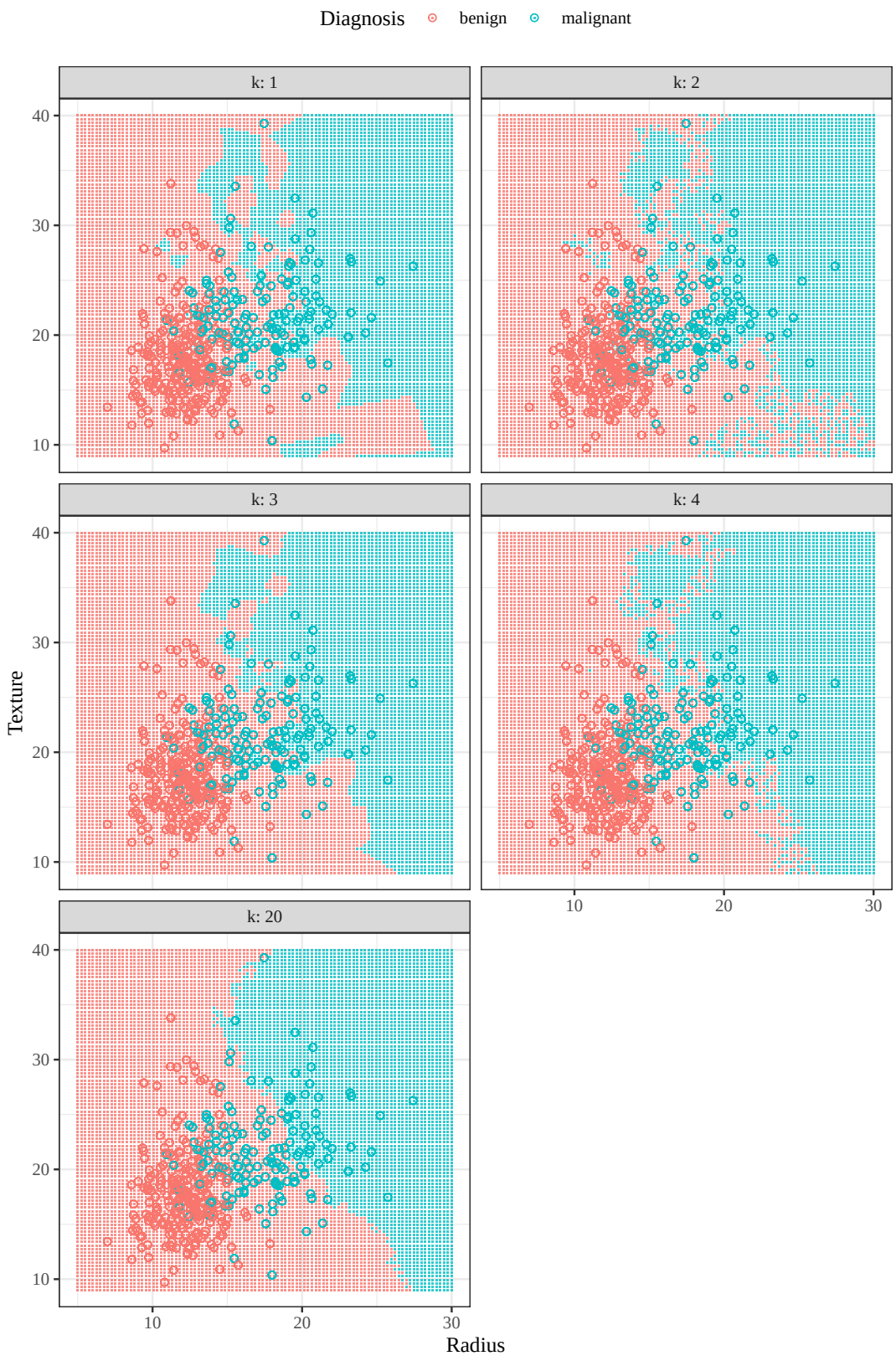
Looking at the training and test confusion matrices for different values of k, we see that the training $k = 1$ model perfectly classifying all observations (because the prediction is based only on the point it was trained on), while larger k generally show more mis-classifications. The test confusion matrices show the opposite trend: with generally fewer classification errors for larger k.

Table 21: kNN predictive accuracy

k	training	test
1	1.000	0.846
2	0.922	0.834
3	0.938	0.888
4	0.915	0.882
20	0.897	0.899

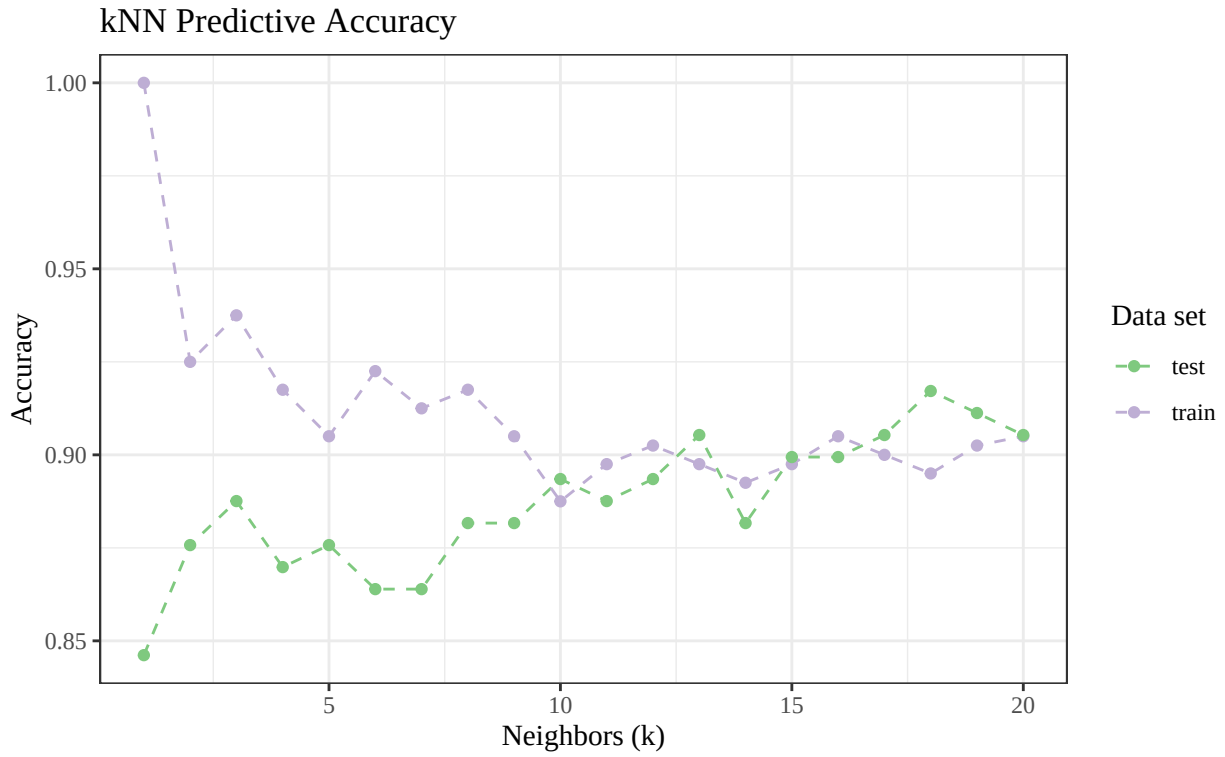
The estimated predictive accuracies tell the same story as the confusion matrices: predictive training accuracy is negatively associated with k, and predictive test accuracy has a positive association.

Part (b)


Decision Boundary of the kNN model
With different k neighbors
Diagnosis
benign
malignant
k: 1
k: 2
k: 3
k: 4
k: 20
Texture
Radius
10
20
30
40

For low k the decision boundary is highly non-linear, closely following the training observations. As k increases, the boundary begins to smooth, fitting the training data less well, but approaching a more believable approximation of the boundary between the true underlying distributions. At the highest k, however, the boundary is informed by so many neighbors that is begins to trend towards the average between the two outcomes.

Part (c)



As k increases, the predictive accuracy on the test data increases, which is indicative of better model performance. As such, I would choose a k between 15 and 20, since in this test-train split of data the predictive test accuracy peaks at $k = 18$.

Question 5

Table 22: Summary of model prediction accuracies

model	threshold/k	train	test
logistic	0.5	0.890	0.905
lda	0.5	0.885	0.888
qda	0.5	0.887	0.882
kNN	1.0	1.000	0.846
kNN	10.0	0.887	0.893
kNN	20.0	0.905	0.905

Overall, The k-nearest neighbors model has the best predictive accuracy, though the test accuracy of the logistic model comes close. Given that the ideal decision boundary of our data will be highly non-linear, we expect the kNN model to perform the best because it is a non-parametric model. The logistic, LDA, and QDA models are all linear, meaning they are fundamentally limited to fitting linear decision boundaries.

One possible mitigation technique to approximate a more nonlinear boundary with linear models is to include more dimensions in the model. This comes with the cost of increased variance, but with the benefit a reduction in bias.

Appendix

Analysis

```
# Prep work -------------------------------------------------------------

library(MASS)
library(class)
library(readr)
library(dplyr)
library(ggplot2)

source("functions/gen_predict_bayes.R")
source("functions/confusion_table.R")
source("functions/plot_decision_boundary.R")
source("functions/calculate_roc.R")

df_wdbc <- read_csv("data/wdbc.data", col_names = FALSE) %>%
  select(diagnosis = 2, radius = 3, texture = 4) %>%
  mutate(
    diagnosis = factor(
      diagnosis, levels = c("B", "M"), labels = c("benign", "malignant")
    )
  )

pred_point <- function(.model, .type = "link", ...) {
  predict(.model, type = .type, newdata = data.frame(...))
}

set.seed(123456)


# Question 1a -----------------------------------------------------------

obs_by_class <- df_wdbc %>%
  group_by(diagnosis) %>%
  summarise(observations = n())


# Question 1b -----------------------------------------------------------

train_obs <- 400

df_shuffled <- df_wdbc[sample(nrow(df_wdbc)), ]
df_train <- df_shuffled[1:train_obs, ]
df_test <- df_shuffled[(train_obs + 1):nrow(df_shuffled), ]


# Question 1c -----------------------------------------------------------

plot_scatter <-
  ggplot(df_train, aes(x = radius, y = texture, color = diagnosis)) +
  geom_point() +
```

```
  theme_bw(base_family = "serif") +
  labs(
    title = "Scatter of Tumor Texture vs Radius",
    subtitle = "Grouped by diagnosis",
    x = "Radius",
    y = "Texture",
    color = "Diagnosis"
  )


# Question 1d ------------------------------------------------------------

model_logistic <- glm(diagnosis ~ ., family = "binomial", data = df_train)

coef_logistic <- coef(model_logistic)


# Question 1e ------------------------------------------------------------

logodds_calc_1e <- sum(coef_logistic * c(1, 10, 12))
prob_calc_1e <- plogis(logodds_calc_1e)
prob_pred_1e <- pred_point(model_logistic, "response", radius = 10, texture = 12)


# Question 1f ------------------------------------------------------------

prob_calc_1f <- exp(.7) / (1 + exp(.7))


# Question 1g ------------------------------------------------------------

pred_bayes_logistic <- gen_predict_bayes(
  function(data) predict(model_logistic, data, type = "response")
)

class_logistic <- list(
  train = pred_bayes_logistic(df_train, .5),
  test = pred_bayes_logistic(df_test, .5)
)

confusion_logistic <- list(
  train = confusion_table(class_logistic$train, diagnosis, pred_class),
  test = confusion_table(class_logistic$test, diagnosis, pred_class)
)

acc_logistic <- list(
  train = with(class_logistic$train, mean(diagnosis == pred_class)),
  test = with(class_logistic$test, mean(diagnosis == pred_class))
)


# Question 1h ------------------------------------------------------------
```

```r
plot_logistic_grid <-
  plot_decision_boundary(df_train, pred_bayes_logistic, c(.25, .5, .75)) +
  labs(
    title = "Decision Boundary of the logistic model",
    subtitle = "With different classification thresholds"
  )


# Question 1i ------------------------------------------------------------

roc_logistic <- calculate_roc(df_test, pred_bayes_logistic, seq(0, 1, by = .01))


# Question 1j ------------------------------------------------------------

auc_logistic <- roc_logistic$AUC$value


# Question 2a ------------------------------------------------------------

model_lda <- lda(diagnosis ~ ., df_train, center = TRUE, scale = TRUE)

summary_lda <- model_lda$means %>%
  as_tibble(rownames = "diagnosis") %>%
  mutate(prior = model_lda$prior) %>%
  select(diagnosis, prior, everything())


# Question 2b ------------------------------------------------------------

pred_bayes_lda <- gen_predict_bayes(
  function(data) predict(model_lda, data)$posterior[, "malignant"]
)

class_lda <- list(
  train = pred_bayes_lda(df_train, .5),
  test = pred_bayes_lda(df_test, .5)
)

confusion_lda <- list(
  train = confusion_table(class_lda$train, diagnosis, pred_class),
  test = confusion_table(class_lda$test, diagnosis, pred_class)
)

acc_lda <- list(
  train = with(class_lda$train, mean(diagnosis == pred_class)),
  test = with(class_lda$test, mean(diagnosis == pred_class))
)


# Question 2c ------------------------------------------------------------

plot_lda_grid <-
```

```r
  plot_decision_boundary(df_train, pred_bayes_lda, c(.25, .5, .75)) +
  labs(
    title = "Decision Boundary of the LDA model",
    subtitle = "With different classification thresholds"
  )


# Question 2d ------------------------------------------------------------

roc_lda <- calculate_roc(df_test, pred_bayes_lda, seq(0, 1, by = .01))


# Question 2e ------------------------------------------------------------

auc_lda <- roc_lda$AUC$value


# Question 3a ------------------------------------------------------------

model_qda <- qda(diagnosis ~ ., df_train, center = TRUE, scale = TRUE)

summary_qda <- model_qda$means %>%
  as_tibble(rownames = "diagnosis") %>%
  mutate(prior = model_qda$prior) %>%
  select(diagnosis, prior, everything())


# Question 3b ------------------------------------------------------------

pred_bayes_qda <- gen_predict_bayes(
  function(data) predict(model_qda, data)$posterior[, "malignant"]
)

class_qda <- list(
  train = pred_bayes_qda(df_train, .5),
  test = pred_bayes_qda(df_test, .5)
)

confusion_qda <- list(
  train = confusion_table(class_qda$train, diagnosis, pred_class),
  test = confusion_table(class_qda$test, diagnosis, pred_class)
)

acc_qda <- list(
  train = with(class_qda$train, mean(diagnosis == pred_class)),
  test = with(class_qda$test, mean(diagnosis == pred_class))
)


# Question 3c ------------------------------------------------------------

plot_qda_grid <-
  plot_decision_boundary(df_train, pred_bayes_qda, c(.25, .5, .75)) +
```

```
  labs(
    title = "Decision Boundary of the QDA model",
    subtitle = "With different classification thresholds"
  )


# Question 3d ------------------------------------------------------------

roc_qda <- calculate_roc(df_test, pred_bayes_qda, seq(0, 1, by = .01))


# Question 3e ------------------------------------------------------------

auc_qda <- roc_qda$AUC$value


# Question 4a ------------------------------------------------------------

pred_knn <- function(data, k_neighbors) {

  df_train_scale <- df_train %>%
    mutate(across(radius:texture, ~as.numeric(scale(.x))))

  df_test_scale <- data %>%
    mutate(across(radius:texture, ~as.numeric(scale(.x))))

  add_preds <- function(k) {
    data %>%
        dplyr::mutate(
          pred_class = knn(
            train = dplyr::select(df_train_scale, radius:texture),
            test = dplyr::select(df_test_scale, radius:texture),
            cl = df_train_scale$diagnosis,
            k = k
          ),
          k = k
        )
  }

  k_neighbors %>% purrr::map_dfr(add_preds)

}


class_knn <- list(
  train = pred_knn(df_train, c(1, 2, 3, 4, 20)) %>% group_by(k),
  test = pred_knn(df_test, c(1, 2, 3, 4, 20)) %>% group_by(k)
)

confusion_knn <- list(
  train = class_knn$train %>% confusion_table(diagnosis, pred_class),
  test = class_knn$test %>% confusion_table(diagnosis, pred_class)
)
```

```r
acc_knn <- list(
  train = class_knn$train %>% summarise(accuracy = mean(diagnosis == pred_class)),
  test = class_knn$test %>% summarise(accuracy = mean(diagnosis == pred_class))
)


# Question 4b ------------------------------------------------------------

plot_knn_grid <-
  plot_decision_boundary(df_train, pred_knn, c(1, 2, 3, 4, 20)) +
  facet_wrap(vars(k), ncol = 2, labeller = "label_both") +
  labs(
    title = "Decision Boundary of the kNN model",
    subtitle = "With different k neighbors"
  )


# Question 4c ------------------------------------------------------------

pred_acc_knn <-
  list(train = df_train, test = df_test) %>%
  purrr::map(~pred_knn(.x, 1:20) %>% group_by(k)) %>%
  purrr::map_dfr(
    ~summarise(.x, accuracy = mean(diagnosis == pred_class)),
    .id = "data_set"
  )

plot_pred_acc_knn <-
  ggplot(pred_acc_knn, aes(x = k, y = accuracy, color = data_set)) +
  geom_line(lty = "dashed") +
  geom_point() +
  scale_color_brewer(type = "qual") +
  theme_bw(base_family = "serif") +
  labs(
    title = "kNN Predictive Accuracy",
    x = "Neighbors (k)",
    y = "Accuracy",
    color = "Data set"
  )
```

Helper Functions

```r
gen_predict_bayes <- function(pred_func) {

  ## Arguments:
  #
  #  pred_func
  #    A function accepting a single nod-default argument `data`, which returns
  #    a vector of probabilities of a malignant diagnosis

  function(data, thresholds) {
```

```
    add_classification <- function(threshold) {

      pred_class <- factor(
        pred_func(data) > threshold,
        levels = c(FALSE, TRUE),
        labels = c("benign", "malignant")
      )

      data %>% dplyr::mutate(
        pred_class = pred_class,
        threshold = threshold
      )

    }

    thresholds %>% purrr::map_dfr(add_classification)

  }

}
```

```
confusion_table <- function(data, obs_class, pred_class) {

  data %>%
    dplyr::group_by({{obs_class}}, {{pred_class}}, .add = TRUE) %>%
    dplyr::summarise(count = n()) %>%
    tidyr::pivot_wider(
      names_from = {{pred_class}},
      values_from = count,
      values_fill = 0
    ) %>%
    dplyr::ungroup({{obs_class}}) %>%
    dplyr::rename(`obs / pred` = {{obs_class}})

}
```

```
plot_decision_boundary <- function(data, pred_func, thresholds) {

  grid_points <- expand.grid(
    radius = seq(5, 30, length.out = 100),
    texture = seq(9, 40, length.out = 100)
  )

  data_rep <- thresholds %>%
    purrr::map_dfr(~dplyr::mutate(data, threshold = .x))

  grid_classes <- pred_func(grid_points, thresholds)

  ggplot(grid_classes, aes(x = radius, y = texture, color = pred_class)) +
    geom_point(shape = ".", alpha = .8) +
    geom_point(data = data_rep, aes(color = diagnosis), shape = 1) +
    facet_wrap(vars(threshold), ncol = 2, labeller = "label_both") +
    theme_bw(base_family = "serif") +
```

```
    theme(legend.position = "top") +
    labs(
      x = "Radius",
      y = "Texture",
      color = "Diagnosis"
    )

}
```

```
calculate_roc <- function(data, pred_func, thresholds) {

  metric_tbl <- function(obs, pred) {

    cf_tbl <- table(obs, pred)

    P <- sum(cf_tbl["malignant",])
    N <- sum(cf_tbl["benign",])
    TP <- cf_tbl["malignant", "malignant"]
    TN <- cf_tbl["benign", "benign"]

    tibble(tpr = TP/P, fpr = 1 - TN/N)


  }

  roc_data <- pred_func(data, thresholds) %>%
    group_by(threshold) %>%
    summarise(metric_tbl(diagnosis, pred_class))

  roc_func <- with(roc_data, approxfun(x = fpr, y = tpr))
  auc <- integrate(roc_func, 0, 1)

  roc_plot <- ggplot(roc_data, aes(x = fpr, y = tpr)) +
    geom_path(color = "red") +
    geom_abline(slope = 1, intercept = 0, lty = "dashed", color = "gray") +
    theme_bw(base_family = "serif") +
    labs(
      title = "ROC Curve",
      subtitle = sprintf("AUC: %0.4f", auc$value),
      x = "False Positive Rate (FPR)",
      y = "True Positive Rate (TPR)"
    )

  list(data = roc_data, plot = roc_plot, AUC = auc)

}
```