Breast Cancer Project Part 3

Set the seed to 1, then create a data partition splitting **brca$y** and the **scaled** version of the **brca$x** matrix into a 20% test set and 80% train using the following code:

```
set.seed(1) # if using R 3.5 or earlier
set.seed(1, sample.kind = "Rounding")    # if using R 3.6 or later
test_index <- createDataPartition(brca$y, times = 1, p = 0.2, list = FALSE)
test_x <- x_scaled[test_index,]
test_y <- brca$y[test_index]
train_x <- x_scaled[-test_index,]
train_y <- brca$y[-test_index]
```

You will be using these training and test sets throughout the exercises in Parts 3 and 4. Save your models as you go, because at the end, you'll be asked to make an ensemble prediction and to compare the accuracy of the various models!

Question 9: Training and test sets

2.0/2.0 points (graded)

Check that the training and test sets have similar proportions of benign and malignant tumors.

What proportion of the training set is benign?

0.628 ✓ **Answer:** 0.628

0.628

Explanation

The portion that is benign can be calculated using the following code:

```
mean(train_y == "B")
```

What proportion of the test set is benign?

0.626

✓ **Answer:** 0.626

0.626

Explanation

The portion that is benign can be calculated using the following code:

```
mean(test_y == "B")
```

Submit

You have used 1 of 10 attempts

ℹ Answers are displayed within the problem

Question 10a: K-means Clustering

1.0/1.0 point (graded)

The `predict_kmeans` function defined here takes two arguments - a matrix of observations `x` and a k-means object `k` - and assigns each row of `x` to a cluster from `k`.

```
predict_kmeans <- function(x, k) {
    centers <- k$centers    # extract cluster centers
    # calculate distance to cluster centers
    distances <- sapply(1:nrow(x), function(i){
                        apply(centers, 1, function(y) dist(rbind(x[i,], y)))
                 })
  max.col(-t(distances))  # select cluster with min distance to center
}
```

Set the seed to 3. Perform k-means clustering on the training set with 2 centers and assign the output to `k`. Then use the `predict_kmeans` function to make predictions on the test set.

What is the overall accuracy?

0.922

✓ **Answer:** 0.922 **or** 0.896

0.922

Explanation

The overall accuracy can be calculated using the following code:

```
set.seed(3) # if using R 3.5 or earlier
set.seed(3, sample.kind = "Rounding")    # if using R 3.6 or later
k <- kmeans(train_x, centers = 2)
kmeans_preds <- ifelse(predict_kmeans(test_x, k) == 1, "B", "M")
mean(kmeans_preds == test_y)
```

Submit You have used 1 of 10 attempts

ℹ Answers are displayed within the problem

Question 10b: K-means Clustering

2.0/2.0 points (graded)

What proportion of benign tumors are correctly identified?

0.986 ✓ **Answer:** 0.986 **or** 0.958

0.986

Explanation

The proportion of benign tumors that are correctly identified, which is the sensitivity for benign tumors, can be found using the following code:

```
sensitivity(factor(kmeans_preds), test_y, positive = "B")
```

What proportion of malignant tumors are correctly identified?

0.814 ✓ **Answer:** 0.814 **or** 0.791

0.814

Explanation

The proportion of malignant tumors that are correctly identified, which is the sensitivity for malignant tumors, can be found using the following code:

```
sensitivity(factor(kmeans_preds), test_y, positive = "M")
```

Submit You have used 2 of 10 attempts

ℹ Answers are displayed within the problem

Question 11: Logistic regression model

1.0/1.0 point (graded)

Fit a logistic regression model on the training set using all predictors. Ignore warnings about the algorithm not converging. Make predictions on the test set.

What is the accuracy of the logistic regression model on the test set?

0.957 ✓ **Answer:** 0.957 **or** 0.939

0.957

Explanation
The accuracy of the logistic regression model can be calculated using the following code:

```
train_glm <- train(train_x, train_y,
                    method = "glm")
glm_preds <- predict(train_glm, test_x)
mean(glm_preds == test_y)
```

Submit You have used 1 of 10 attempts

Answers are displayed within the problem

Question 12: LDA and QDA models

2.0/2.0 points (graded)
Train an LDA model and a QDA model on the training set. Make predictions on the test set using each model.

What is the accuracy of the LDA model on the test set?

0.991 ✓ **Answer:** 0.974 **or** 0.991

0.991

Explanation
The accuracy can be determined using the following code:

```
train_lda <- train(train_x, train_y,
                    method = "lda")
lda_preds <- predict(train_lda, test_x)
mean(lda_preds == test_y)
```

What is the accuracy of the QDA model on the test set?

0.957 ✓ **Answer:** 0.948 **or** 0.957

0.957

Explanation

The accuracy can be determined using the following code:

```
train_qda <- train(train_x, train_y,
                   method = "qda")
qda_preds <- predict(train_qda, test_x)
mean(qda_preds == test_y)
```

Submit You have used 1 of 10 attempts

ℹ Answers are displayed within the problem

Question 13: Loess model

1.0/1.0 point (graded)

Set the seed to 5, then fit a loess model on the training set with the `caret` package. You will need to install the `gam` package if you have not yet done so. Use the default tuning grid. This may take several minutes; ignore warnings. Generate predictions on the test set.

What is the accuracy of the loess model on the test set?

0.983 ✓ **Answer:** 0.93 **or** 0.983

0.983

Explanation

The accuracy can be determined using the following code:

```
set.seed(5)
# set.seed(5, sample.kind = "Rounding")    # simulate R 3.5
train_loess <- train(train_x, train_y,
                     method = "gamLoess")
loess_preds <- predict(train_loess, test_x)
mean(loess_preds == test_y)
```

Submit You have used 1 of 10 attempts

ℹ Answers are displayed within the problem