```
> set.seed(1, sample.kind = "Rounding")
Warning message:
In set.seed(1, sample.kind = "Rounding") :
  non-uniform 'Rounding' sampler used
> 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$y[-test_index]
>
>
>
> mean(train_y == "B")
[1] 0.628
> mean(test_y == "B")
[1] 0.626
>
>
>
> predict_kmeans <- function(x, k) {
+     centers <- k$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.seed(3, sample.kind = "Rounding")
Warning message:
In set.seed(3, sample.kind = "Rounding") :
  non-uniform 'Rounding' sampler used
> k <- kmeans(train_x, centers = 2)
> kmeans_preds <- ifelse(predict_kmeans(test_x, k) == 1, "B", "M")
> mean(kmeans_preds == test_y)
[1] 0.922
>
>
>
> sensitivity(factor(kmeans_preds), test_y, positive = "B")
[1] 0.986
> sensitivity(factor(kmeans_preds), test_y, positive = "M")
[1] 0.814
>
>
>
> train_glm <- train(train_x, train_y,
+                      method = "glm")
There were 50 or more warnings (use warnings() to see the first 50)
> glm_preds <- predict(train_glm, test_x)
> mean(glm_preds == test_y)
[1] 0.957
>
>
>
> train_lda <- train(train_x, train_y,
+                      method = "lda")
> lda_preds <- predict(train_lda, test_x)
> mean(lda_preds == test_y)
[1] 0.991
>
> train_qda <- train(train_x, train_y,
+                      method = "qda")
> qda_preds <- predict(train_qda, test_x)
> mean(qda_preds == test_y)
[1] 0.957
```

```
>
>
>
> set.seed(5, sample.kind = "Rounding")
Warning message:
In set.seed(5, sample.kind = "Rounding") :
  non-uniform 'Rounding' sampler used
> train_loess <- train(train_x, train_y,
+                      method = "gamLoess")
Loading required package: gam
Loading required package: splines
Loading required package: foreach

Attaching package: ‘foreach’

The following objects are masked from ‘package:purrr’:

    accumulate, when

Loaded gam 1.16.1

There were 50 or more warnings (use warnings() to see the first 50)
> loess_preds <- predict(train_loess, test_x)
There were 45 warnings (use warnings() to see them)
> mean(loess_preds == test_y)
[1] 0.983
>
>
>
> set.seed(7, sample.kind = "Rounding")
Warning message:
In set.seed(7, sample.kind = "Rounding") :
  non-uniform 'Rounding' sampler used
> tuning <- data.frame(k = seq(3, 21, 2))
> train_knn <- train(train_x, train_y,
+       method = "knn",
+       tuneGrid = tuning)
> train_knn$bestTune
    k
10 21
>
> knn_preds <- predict(train_knn, test_x)
> mean(knn_preds == test_y)
[1] 0.948
>
>
>
> set.seed(9, sample.kind = "Rounding")
Warning message:
In set.seed(9, sample.kind = "Rounding") :
  non-uniform 'Rounding' sampler used
> tuning <- data.frame(mtry = c(3, 5, 7, 9))
> train_rf <- train(train_x, train_y,
+                    method = "rf",
+                    tuneGrid = tuning,
+                    importance = TRUE)
> train_rf$bestTune
  mtry
1    3
>
> rf_preds <- predict(train_rf, test_x)
> mean(rf_preds == test_y)
[1] 0.974
>
> varImp(train_rf)
```

```
rf variable importance

  only 20 most important variables shown (out of 30)

                  Importance
area_worst             100.0
radius_worst            87.7
concave_pts_worst       85.7
perimeter_worst         85.5
concave_pts_mean        72.1
area_se                 67.3
concavity_worst         63.5
area_mean               61.4
texture_worst           59.9
perimeter_mean          55.2
concavity_mean          55.2
texture_mean            55.0
radius_se               49.8
smoothness_worst        49.1
radius_mean             49.0
perimeter_se            45.0
compactness_worst       39.3
symmetry_worst          35.3
smoothness_mean         30.6
fractal_dim_worst       27.8
>
>
>
> ensemble <- cbind(glm = glm_preds == "B", lda = lda_preds == "B", qda = qda_preds == "B", loess = loess_
preds == "B", rf = rf_preds == "B", knn = knn_preds == "B", kmeans = kmeans_preds == "B")
>
> ensemble_preds <- ifelse(rowMeans(ensemble) > 0.5, "B", "M")
> mean(ensemble_preds == test_y)
[1] 0.983
>
>
>
> models <- c("K means", "Logistic regression", "LDA", "QDA", "Loess", "K nearest neighbors", "Random fore
st", "Ensemble")
> accuracy <- c(mean(kmeans_preds == test_y),
+               mean(glm_preds == test_y),
+               mean(lda_preds == test_y),
+               mean(qda_preds == test_y),
+               mean(loess_preds == test_y),
+               mean(knn_preds == test_y),
+               mean(rf_preds == test_y),
+               mean(ensemble_preds == test_y))
> data.frame(Model = models, Accuracy = accuracy)
                Model Accuracy
1             K means    0.922
2 Logistic regression    0.957
3                 LDA    0.991
4                 QDA    0.957
5               Loess    0.983
6 K nearest neighbors    0.948
7       Random forest    0.974
8            Ensemble    0.983
>
>
```