## **Breast Cancer**

Elena Knudsen

5/28/2021

data(brca)

#Using the brca dataset from the dslabs package contains informatuon about breast cancer diagnosis biopsy samples for tumors determined to be malignant or benign #Load data and libraries options(digits = 3) library(matrixStats) library(tidyverse)

- tidyverse 1.3.0 -

## — Attaching packages ## ✓ ggplot2 3.3.2 ✓ purrr 0.3.4 ## / tibble 3.0.3 / dplyr 1.0.2 ## ✓ tidyr 1.1.0 ✓ stringr 1.4.0

## / readr 1.3.1 / forcats 0.5.0 ## -- Conflicts tidyverse\_conflicts() — ## x dplyr::count() masks matrixStats::count() ## x dplyr::filter() masks stats::filter() ## x dplyr::lag() masks stats::lag()

library(caret)

## Loading required package: lattice

## Attaching package: 'caret' ## The following object is masked from 'package:purrr':

## lift library(dslabs)

#Number of samples within dataset dim(brca\$x)[1] ## [1] 569

#Proportion of malignant samples mean(brca\$y == "M")

**##** [1] 0.373

#Scaling matrix x\_centered <- sweep(brca\$x, 2, colMeans(brca\$x))</pre> x\_scaled <- sweep(x\_centered, 2, colSds(brca\$x), FUN = "/")</pre> sd(x\_scaled[,1])

## [1] 1 #Produce heat map of relationship between features using the scaled matrix d\_features <- dist(t(x\_scaled))</pre> heatmap(as.matrix(d\_features), labRow = NA, labCol = NA)

#Principal component analysis of scaled matrix pca <- prcomp(x\_scaled)</pre> summary(pca) ## Importance of components: PC1 PC2 PC3 PC4 PC5 PC7 PC6 ## Standard deviation 3.644 2.386 1.6787 1.407 1.284 1.0988 0.8217 0.6904 ## Proportion of Variance 0.443 0.190 0.0939 0.066 0.055 0.0403 0.0225 0.0159 ## Cumulative Proportion 0.443 0.632 0.7264 0.792 0.847 0.8876 0.9101 0.9260 PC9 PC10 PC11 PC12 PC13 PC14 PC15 0.6457 0.5922 0.5421 0.51104 0.49128 0.39624 0.30681 ## Standard deviation ## Proportion of Variance 0.0139 0.0117 0.0098 0.00871 0.00805 0.00523 0.00314 ## Cumulative Proportion 0.9399 0.9516 0.9614 0.97007 0.97812 0.98335 0.98649 PC16 PC17 PC18 PC19 PC20 PC21 ## Standard deviation 0.28260 0.24372 0.22939 0.22244 0.17652 0.173 0.16565 ## Proportion of Variance 0.00266 0.00198 0.00175 0.00165 0.00104 0.001 0.00091 ## Cumulative Proportion 0.98915 0.99113 0.99288 0.99453 0.99557 0.997 0.99749 PC23 PC24 PC25 PC26 PC27 PC28 ## Standard deviation 0.15602 0.1344 0.12442 0.09043 0.08307 0.03987 0.02736 ## Proportion of Variance 0.00081 0.0006 0.00052 0.00027 0.00023 0.00005 0.00002 ## Cumulative Proportion 0.99830 0.9989 0.99942 0.99969 0.99992 0.99997 1.00000 PC30 ## Standard deviation 0.0115 ## Proportion of Variance 0.0000

## Cumulative Proportion 1.0000

train\_y <- brca\$y[-test\_index]</pre>

mean(train\_y == "B")

## [1] 0.628

0.628

0.896

## [1] 0.896

## [1] 0.958

## [1] 0.958

0.958

#Proportion of benign tumors correctly identified

## Warning: glm.fit: algorithm did not converge

#K-nearest neighbors model

tuning  $\leftarrow$  data.frame(k = seq(3, 21, 2))

tuning  $\leftarrow$  data.frame(mtry = c(3, 5, 7, 9))

method = "rf",

tuneGrid = tuning, importance = TRUE)

train\_lda <- train(train\_x, train\_y, method = "lda")</pre>

train\_qda <- train(train\_x, train\_y, method = "qda")</pre>

lda\_preds <- predict(train\_lda, test\_x)</pre>

qda\_preds <- predict(train\_qda, test\_x)</pre>

mean(lda\_preds == test\_y)

mean(qda\_preds == test\_y)

## 5 K nearest neighbors

Random forest

Ensemble

## 6

## 7

0.974

0.948

0.965

## [1] 0.974

**#QDA** model accuracy

train\_rf <- train(train\_x, train\_y,</pre>

method = "knn",

tuneGrid = tuning)

train\_knn <- train(train\_x, train\_y,</pre>

set.seed(7)

set.seed(9)

train\_rf\$bestTune

train knn\$bestTune

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

sensitivity(factor(kmeans\_preds), test\_y, positive = "B")

#Proportion of training set benign

geom\_point()

data.frame(pca\$x[,1:2], type = brca\$y) %>% ggplot(aes(PC1, PC2, color = type)) +

#Plot two principal components with color representing tumor type (M,B)

type -5 **-**-10 **-**-5 10 PC1 #Training and test sets set.seed(1) test\_index <- createDataPartition(brca\$y, times = 1, p = 0.2, list = FALSE)</pre> test\_x <- x\_scaled[test\_index,]</pre> test\_y <- brca\$y[test\_index]</pre> train\_x <- x\_scaled[-test\_index,]</pre>

## [1] 0.628 #Test set mean(test\_y == "B") ## [1] 0.626 0.626 ## [1] 0.626 #Using predict\_kmeans() to make predictions on test set predict\_kmeans <- function(x, k) {</pre> centers <- k\$centers distances <- sapply(1:nrow(x), function(i){</pre> apply(centers, 1, function(y) dist(rbind(x[i,], y))) max.col(-t(distances)) # select cluster with min distance to center #Overall accuracy set.seed(3) k <- kmeans(train\_x, centers = 2)</pre> kmeans\_preds <- ifelse(predict\_kmeans(test\_x, k) == 1, "B", "M")</pre> mean(kmeans\_preds == test\_y) ## [1] 0.896

#Proportion of malignant tumors correctly identified sensitivity(factor(kmeans\_preds), test\_y, positive = "M") ## [1] 0.791 0.791 ## [1] 0.791 #Testing accuracy of logistic regression model train\_glm <- train(train\_x, train\_y, method = "glm")</pre> ## Warning: glm.fit: algorithm did not converge ## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred ## Warning: glm.fit: algorithm did not converge ## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge ## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred ## Warning: glm.fit: algorithm did not converge ## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred ## Warning: glm.fit: algorithm did not converge ## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred ## Warning: glm.fit: algorithm did not converge ## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred ## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge ## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred ## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: algorithm did not converge ## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred ## Warning: glm.fit: algorithm did not converge ## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge ## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred ## Warning: glm.fit: algorithm did not converge ## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred glm\_preds <- predict(train\_glm, test\_x)</pre> mean(glm preds == test y) ## [1] 0.939

## k ## 4 9 knn\_preds <- predict(train\_knn, test\_x)</pre> mean(knn\_preds == test\_y) ## [1] 0.974 #Random forest model accuracy

## mtry ## 1 3 rf\_preds <- predict(train\_rf, test\_x)</pre> mean(rf\_preds == test\_y) ## [1] 0.948 #LDA model accuracy

## [1] 0.948 #Create ensemble to generate majority prediction of tumor type ensemble <- cbind(glm = glm\_preds == "B", lda = lda\_preds == "B", qda = qda\_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] 0.965

#Calculate most accurate model models <- c("K means", "Logistic regression", "lda", "QDA", "K nearest neighbors", "Random forest", "Ensemble")</pre> accuracy <- c(mean(kmeans\_preds == test\_y),</pre> mean(glm\_preds == test\_y), mean(lda\_preds == test\_y), mean(qda\_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 0.896 ## 2 Logistic regression 0.939 ## 3 0.974 ## 4 QDA 0.948