Project Data Mining

Dana Abdirakhym

2024-03-18

Load libraries

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.3.1
```

```
## Loading required package: lattice
```

```
library(tidyverse) # for data manipulation and visualization
```

```
## Warning: package 'readr' was built under R version 4.3.1
```

```
## Warning: package 'dplyr' was built under R version 4.3.1
```

```
## Warning: package 'stringr' was built under R version 4.3.1
```

```
## Warning: package 'lubridate' was built under R version 4.3.1
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(randomForest)
```

```
## randomForest 4.7-1.1
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:dplyr':
##
##     combine
##
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(e1071) # for SVM
```

```
## Warning: package 'e1071' was built under R version 4.3.1
```

```
library(nnet) # for neural network
install.packages("kernlab")
```

```
##
## The downloaded binary packages are in
##  /var/folders/zw/9p94jsvn2mb68h0wkh7y2zx80000gn/T//RtmpLok4Oy/downloaded_packages
```

```
library(kernlab)
```

```
##
## Attaching package: 'kernlab'
##
## The following object is masked from 'package:purrr':
##
##     cross
##
## The following object is masked from 'package:ggplot2':
##
##     alpha
```

Step 1. Import and parse breastcancer data set

```
library(readr)
data <- read_csv("/Users/danaabdirakhym/Downloads/breast_cancer.csv")
```

```
## Rows: 569 Columns: 32
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): diagnosis
## dbl (31): id, radius_mean, texture_mean, perimeter_mean, area_mean, smoothne...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
##Coverted Malignant to 1, Benign  to 0
data$diagnosis <- as.factor(ifelse(data$diagnosis == "M", 1, 0))
breast_cancer <- na.omit(data)

breast_cancer <- subset(breast_cancer, select = -id)
colnames(breast_cancer)
```

```
##  [1] "diagnosis"               "radius_mean"
##  [3] "texture_mean"            "perimeter_mean"
##  [5] "area_mean"               "smoothness_mean"
##  [7] "compactness_mean"        "concavity_mean"
##  [9] "concave points_mean"     "symmetry_mean"
## [11] "fractal_dimension_mean"  "radius_se"
## [13] "texture_se"              "perimeter_se"
## [15] "area_se"                 "smoothness_se"
## [17] "compactness_se"          "concavity_se"
## [19] "concave points_se"       "symmetry_se"
## [21] "fractal_dimension_se"    "radius_worst"
## [23] "texture_worst"           "perimeter_worst"
## [25] "area_worst"              "smoothness_worst"
## [27] "compactness_worst"       "concavity_worst"
## [29] "concave points_worst"    "symmetry_worst"
## [31] "fractal_dimension_worst"
```

###Step 2. Create at least four different classifiers. Determine the needed features and create training and test data.

```r
library(caret)
set.seed(123) # for reproducibility

### Splitting to train and test data 80 to 20
index <- createDataPartition(breast_cancer$diagnosis, p=0.8, list=FALSE)
train_data <- breast_cancer[index, ]
test_data <- breast_cancer[-index, ]


head(train_data)
```

```
## # A tibble: 6 x 31
##   diagnosis radius_mean texture_mean perimeter_mean area_mean smoothness_mean
##   <fct>           <dbl>        <dbl>          <dbl>     <dbl>           <dbl>
## 1 1                18.0         10.4          123.      1001           0.118
## 2 1                20.6         17.8          133.      1326           0.0847
## 3 1                19.7         21.2          130       1203           0.110
## 4 1                11.4         20.4           77.6      386.          0.142
## 5 1                20.3         14.3          135.      1297           0.100
## 6 1                12.4         15.7           82.6      477.          0.128
## # i 25 more variables: compactness_mean <dbl>, concavity_mean <dbl>,
## #   `concave points_mean` <dbl>, symmetry_mean <dbl>,
## #   fractal_dimension_mean <dbl>, radius_se <dbl>, texture_se <dbl>,
## #   perimeter_se <dbl>, area_se <dbl>, smoothness_se <dbl>,
## #   compactness_se <dbl>, concavity_se <dbl>, `concave points_se` <dbl>,
```

```
## #   symmetry_se <dbl>, fractal_dimension_se <dbl>, radius_worst <dbl>,
## #   texture_worst <dbl>, perimeter_worst <dbl>, area_worst <dbl>, ...
```

```
head(test_data)
```

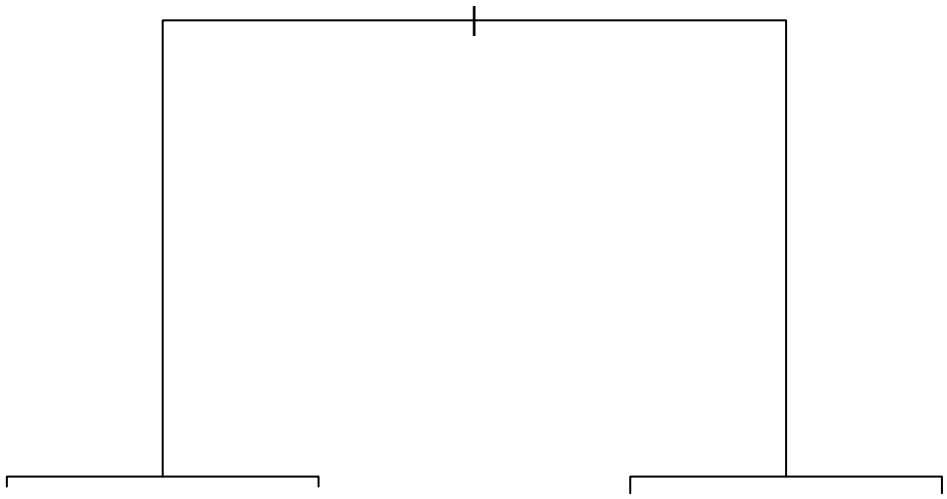
```
## # A tibble: 6 x 31
##   diagnosis radius_mean texture_mean perimeter_mean area_mean smoothness_mean
##   <fct>           <dbl>        <dbl>          <dbl>     <dbl>           <dbl>
## 1 1               19.2          24.8          132.      1123           0.0974
## 2 1               15.8          24.0          104.       783.          0.0840
## 3 1               13.7          22.6           93.6      578.          0.113
## 4 1               14.7          20.1           94.7      684.          0.0987
## 5 0                9.50         12.4           60.3      274.          0.102
## 6 1               17.1          16.4          116        913.          0.119
## # i 25 more variables: compactness_mean <dbl>, concavity_mean <dbl>,
## #   `concave points_mean` <dbl>, symmetry_mean <dbl>,
## #   fractal_dimension_mean <dbl>, radius_se <dbl>, texture_se <dbl>,
## #   perimeter_se <dbl>, area_se <dbl>, smoothness_se <dbl>,
## #   compactness_se <dbl>, concavity_se <dbl>, `concave points_se` <dbl>,
## #   symmetry_se <dbl>, fractal_dimension_se <dbl>, radius_worst <dbl>,
## #   texture_worst <dbl>, perimeter_worst <dbl>, area_worst <dbl>, ...
```

```
# Check the levels before conversion
levels(breast_cancer$diagnosis)
```

```
## [1] "0" "1"
```

```
# Convert factor levels "0" and "1" to actual numeric values 0 and 1
# This conversion ensures that 'diagnosis' is now a numeric variable instead of a factor.
breast_cancer$diagnosis <- as.numeric(as.character(breast_cancer$diagnosis))
train_data$diagnosis <- as.numeric(as.character(train_data$diagnosis))
test_data$diagnosis <- as.numeric(as.character(test_data$diagnosis))
```

```
###Creating the model using recursive partitioning

if (!requireNamespace("rpart", quietly = TRUE)) {
    install.packages("rpart")
}
library(rpart)


# Fit a decision tree model 1

model1_rpart <- rpart(diagnosis ~ ., data = train_data, method = "class")
model1_pred <- predict(model1_rpart, type="class", newdata= test_data)
model1_prob <-predict(model1_rpart, type="prob", newdata= test_data)


plot(model1_rpart, main= "decision tree model 1")
```




```
##create model using conditional inferences
if (!requireNamespace("party", quietly = TRUE)) {
    install.packages("party")
}
library(party)
```

```
## Warning: package 'party' was built under R version 4.3.1
```

```
## Loading required package: grid
```

```
## Loading required package: mvtnorm
```

```
## Warning: package 'mvtnorm' was built under R version 4.3.1
```

```
## Loading required package: modeltools
```

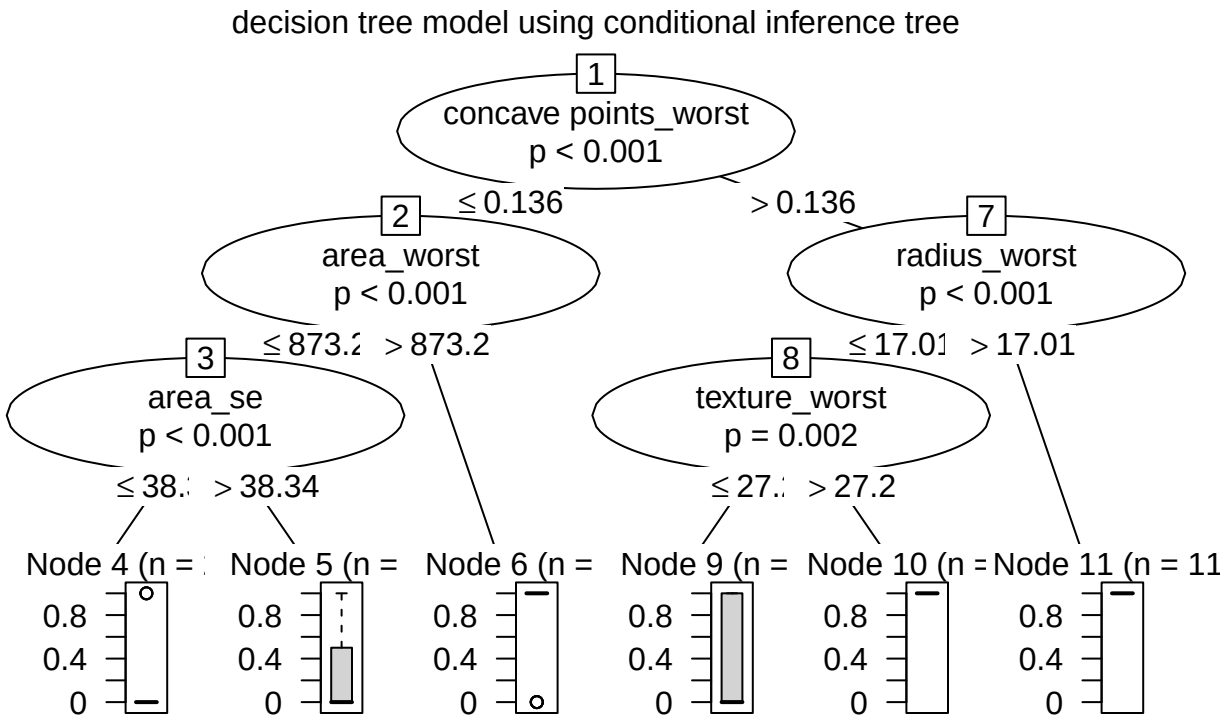
```
## Loading required package: stats4
```

```
##
## Attaching package: 'modeltools'
```

```
## The following object is masked from 'package:kernlab':
##
##     prior
```

```
## Loading required package: strucchange

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

##
## Attaching package: 'strucchange'

## The following object is masked from 'package:stringr':
##
##     boundary

##
## Attaching package: 'party'

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

```r
model2 <- ctree(diagnosis ~ ., data = train_data)
# Predict class labels
model2_pred <- predict(model2, newdata = test_data, type = "response")
# Predict probabilities
model2_prob <- predict(model2, newdata = test_data, type = "prob")

plot(model2, main= "decision tree model using conditional inference tree")
```




```
###create random forest using conditional inference tree

if (!requireNamespace("ipred", quietly = TRUE)) {
    install.packages("ipred")
}
library(ipred)
# Fit a Bagging model
model3_bagging <- bagging(diagnosis ~ ., data = train_data, nbagg = 25)  # 'nbagg' denotes the number o
model3_bagging_pred <- predict(model3_bagging, newdata = test_data, type = "class")
model3_bagging_pred <- predict(model3_bagging, newdata = test_data, type = "prob")
```

##Step 3 Provide your performance measure: confusion matrix, accuracy, recall, etc.

```
if (!requireNamespace("caret", quietly = TRUE)) {
    install.packages("caret")
}
library(caret)

if (!requireNamespace("e1071", quietly = TRUE)) {  # Required for SVM
    install.packages("e1071")
}
if (!requireNamespace("class", quietly = TRUE)) {  # Might be required for KNN
    install.packages("class")
}
# Convert 'diagnosis' to a factor with two levels
```

```r
train_data$diagnosis <- factor(train_data$diagnosis, levels = c('0', '1'))
test_data$diagnosis <- factor(test_data$diagnosis, levels = c('0', '1'))

# Support vector machines
model_svm <- train(diagnosis ~ ., data = train_data, method = "svmRadial")
model_svm_pred <- predict(model_svm, newdata = test_data)
model_svm_prob <- predict(model_svm, newdata = test_data, type = "prob")
```

```
## Warning in method$prob(modelFit = modelFit, newdata = newdata, submodels =
## param): kernlab class probability calculations failed; returning NAs
```

```r
#Knn model
model_knn <- train(diagnosis ~ ., data = train_data, method = "knn")
model_knn_pred <- predict(model_knn, newdata = test_data)
model_knn_prob <- predict(model_knn, newdata = test_data, type = "prob")

#Logistic regression model
model_log <- train(diagnosis ~ ., data = train_data, method = "glm", family = "binomial")
```

```
## 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: 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: 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: 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: 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: algorithm did not converge

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

```
model_log_pred <- predict(model_log, newdata = test_data)
model_log_prob <- predict(model_log, newdata = test_data, type = "prob")


# Calculate performance metrics for the SVM model
conf_matrix_svm <- confusionMatrix(factor(model_svm_pred, levels = levels(test_data$diagnosis)), test_da
precision_svm <- conf_matrix_svm$byClass['Pos Pred Value']  # This is the precision
recall_svm <- conf_matrix_svm$byClass['Sensitivity']  # This is the recall
accuracy_svm <- conf_matrix_svm$overall['Accuracy']

# Calculate performance metrics for the KNN model
conf_matrix_knn <- confusionMatrix(factor(model_knn_pred, levels = levels(test_data$diagnosis)), test_da
precision_knn <- conf_matrix_knn$byClass['Pos Pred Value']
recall_knn <- conf_matrix_knn$byClass['Sensitivity']
accuracy_knn <- conf_matrix_knn$overall['Accuracy']

# Calculate performance metrics for the Logistic Regression model
conf_matrix_log <- confusionMatrix(factor(model_log_pred, levels = levels(test_data$diagnosis)), test_da
precision_log <- conf_matrix_log$byClass['Pos Pred Value']
recall_log <- conf_matrix_log$byClass['Sensitivity']
accuracy_log <- conf_matrix_log$overall['Accuracy']

# Print out the metrics
cat("SVM Metrics: Accuracy =", accuracy_svm, ", Precision =", precision_svm, ", Recall =", recall_svm,
```

```
## SVM Metrics: Accuracy = 0.9823009 , Precision = 1 , Recall = 0.971831
```

```r
cat("KNN Metrics: Accuracy =", accuracy_knn, ", Precision =", precision_knn, ", Recall =", recall_knn,
```

```
## KNN Metrics: Accuracy = 0.9115044 , Precision = 0.942029 , Recall = 0.915493
```

```r
cat("Logistic Regression Metrics: Accuracy =", accuracy_log, ", Precision =", precision_log, ", Recall =
```

```
## Logistic Regression Metrics: Accuracy = 0.9380531 , Precision = 1 , Recall = 0.9014085
```

###Step 4 Combine the classifiers in an ensemble

```r
classifier1_svm <- as.numeric(model_svm_pred) - 1  # Convert factors to numeric
classifier2_knn <- as.numeric(model_knn_pred) - 1
classifier3_log <- as.numeric(model_log_pred) - 1
classifier4_dtree <- as.numeric(model1_pred) - 1

# Assuming your factor levels are correctly ordered, like c('0', '1')
# If they are not, replace '0' and '1' with the actual factor levels from your predictions
classifier1_svm <- ifelse(model_svm_pred == '1', 1, 0)  # Adjust based on your actual levels
classifier2_knn <- ifelse(model_knn_pred == '1', 1, 0)
classifier3_log <- ifelse(model_log_pred == '1', 1, 0)
classifier4_dtree <- ifelse(model1_pred == '1', 1, 0)



combine.df <- data.frame(classifier1_svm, classifier2_knn, classifier3_log, classifier4_dtree)
combine.df
```

```
##     classifier1_svm classifier2_knn classifier3_log classifier4_dtree
## 1                 1               1               1                 1
## 2                 1               1               1                 1
## 3                 1               0               1                 1
## 4                 1               1               1                 1
## 5                 0               0               0                 0
## 6                 1               1               1                 1
## 7                 1               1               1                 0
## 8                 1               0               1                 1
## 9                 1               1               1                 1
## 10                0               0               0                 0
## 11                0               0               0                 0
## 12                0               0               0                 0
## 13                0               0               0                 0
## 14                1               0               1                 1
## 15                1               1               1                 1
## 16                1               1               1                 1
## 17                0               0               0                 0
## 18                1               1               1                 1
## 19                1               0               1                 1
## 20                1               1               1                 1
## 21                1               1               1                 1
## 22                1               1               1                 1
## 23                1               1               1                 1
## 24                1               1               1                 1
```

```
## 25  0  1  1  0
## 26  0  0  0  0
## 27  0  0  0  0
## 28  0  0  0  0
## 29  1  0  1  1
## 30  0  0  0  1
## 31  0  0  0  0
## 32  1  1  1  0
## 33  1  1  1  1
## 34  0  0  0  0
## 35  1  1  1  1
## 36  1  1  1  0
## 37  0  0  0  0
## 38  1  1  1  1
## 39  1  1  1  0
## 40  0  0  0  0
## 41  0  0  1  0
## 42  0  0  0  0
## 43  1  1  1  1
## 44  1  1  1  1
## 45  0  0  0  0
## 46  0  0  0  1
## 47  1  1  1  1
## 48  1  1  1  1
## 49  1  1  1  1
## 50  0  0  0  0
## 51  0  0  0  0
## 52  0  0  0  0
## 53  1  1  1  1
## 54  0  0  0  0
## 55  0  0  1  0
## 56  1  1  1  1
## 57  1  1  1  1
## 58  0  0  0  0
## 59  0  0  1  0
## 60  0  0  0  0
## 61  0  0  0  0
## 62  0  0  0  0
## 63  0  0  0  0
## 64  0  0  0  0
## 65  1  1  1  1
## 66  0  0  0  0
## 67  1  1  1  1
## 68  0  0  1  0
## 69  0  0  0  0
## 70  0  0  0  0
## 71  1  1  1  1
## 72  0  0  0  0
## 73  0  0  0  0
## 74  1  1  1  1
## 75  0  0  0  0
## 76  0  1  0  0
## 77  0  0  0  0
## 78  1  1  1  1
```

```
## 79                0                0                0                0
## 80                0                0                0                0
## 81                0                0                0                0
## 82                0                1                0                0
## 83                1                1                1                1
## 84                0                0                0                0
## 85                1                0                1                1
## 86                0                0                0                0
## 87                0                0                0                0
## 88                0                0                0                1
## 89                1                1                1                1
## 90                0                0                0                0
## 91                1                1                1                1
## 92                0                0                0                0
## 93                0                0                0                0
## 94                0                1                0                1
## 95                0                0                0                0
## 96                0                0                0                1
## 97                0                0                0                0
## 98                0                0                0                1
## 99                0                0                0                0
## 100               0                1                0                0
## 101               1                1                1                1
## 102               0                0                0                0
## 103               0                0                0                0
## 104               0                0                0                0
## 105               0                0                0                0
## 106               0                1                0                0
## 107               0                0                0                0
## 108               0                0                0                1
## 109               0                0                0                0
## 110               0                0                0                0
## 111               0                0                0                0
## 112               0                0                0                0
## 113               1                1                1                1
```

```
combine.df$vote <- rowSums(combine.df)
combine.df$class <- ifelse(combine.df$vote >= 2,1,0)
 # Majority is 'malignant' if 2 or more classifiers predict 1
combine.df
```

```
##     classifier1_svm classifier2_knn classifier3_log classifier4_dtree vote
## 1                 1                1                1                 1    4
## 2                 1                1                1                 1    4
## 3                 1                0                1                 1    3
## 4                 1                1                1                 1    4
## 5                 0                0                0                 0    0
## 6                 1                1                1                 1    4
## 7                 1                1                1                 0    3
## 8                 1                0                1                 1    3
## 9                 1                1                1                 1    4
## 10                0                0                0                 0    0
## 11                0                0                0                 0    0
## 12                0                0                0                 0    0
```

```
## 13  0  0  0  0  0
## 14  1  0  1  1  3
## 15  1  1  1  1  4
## 16  1  1  1  1  4
## 17  0  0  0  0  0
## 18  1  1  1  1  4
## 19  1  0  1  1  3
## 20  1  1  1  1  4
## 21  1  1  1  1  4
## 22  1  1  1  1  4
## 23  1  1  1  1  4
## 24  1  1  1  1  4
## 25  0  1  1  0  2
## 26  0  0  0  0  0
## 27  0  0  0  0  0
## 28  0  0  0  0  0
## 29  1  0  1  1  3
## 30  0  0  0  1  1
## 31  0  0  0  0  0
## 32  1  1  1  0  3
## 33  1  1  1  1  4
## 34  0  0  0  0  0
## 35  1  1  1  1  4
## 36  1  1  1  0  3
## 37  0  0  0  0  0
## 38  1  1  1  1  4
## 39  1  1  1  0  3
## 40  0  0  0  0  0
## 41  0  0  1  0  1
## 42  0  0  0  0  0
## 43  1  1  1  1  4
## 44  1  1  1  1  4
## 45  0  0  0  0  0
## 46  0  0  0  1  1
## 47  1  1  1  1  4
## 48  1  1  1  1  4
## 49  1  1  1  1  4
## 50  0  0  0  0  0
## 51  0  0  0  0  0
## 52  0  0  0  0  0
## 53  1  1  1  1  4
## 54  0  0  0  0  0
## 55  0  0  1  0  1
## 56  1  1  1  1  4
## 57  1  1  1  1  4
## 58  0  0  0  0  0
## 59  0  0  1  0  1
## 60  0  0  0  0  0
## 61  0  0  0  0  0
## 62  0  0  0  0  0
## 63  0  0  0  0  0
## 64  0  0  0  0  0
## 65  1  1  1  1  4
## 66  0  0  0  0  0
```

```
## 67                1                1                1                1    4
## 68                0                0                1                0    1
## 69                0                0                0                0    0
## 70                0                0                0                0    0
## 71                1                1                1                1    4
## 72                0                0                0                0    0
## 73                0                0                0                0    0
## 74                1                1                1                1    4
## 75                0                0                0                0    0
## 76                0                1                0                0    1
## 77                0                0                0                0    0
## 78                1                1                1                1    4
## 79                0                0                0                0    0
## 80                0                0                0                0    0
## 81                0                0                0                0    0
## 82                0                1                0                0    1
## 83                1                1                1                1    4
## 84                0                0                0                0    0
## 85                1                0                1                1    3
## 86                0                0                0                0    0
## 87                0                0                0                0    0
## 88                0                0                0                1    1
## 89                1                1                1                1    4
## 90                0                0                0                0    0
## 91                1                1                1                1    4
## 92                0                0                0                0    0
## 93                0                0                0                0    0
## 94                0                1                0                1    2
## 95                0                0                0                0    0
## 96                0                0                0                1    1
## 97                0                0                0                0    0
## 98                0                0                0                1    1
## 99                0                0                0                0    0
## 100               0                1                0                0    1
## 101               1                1                1                1    4
## 102               0                0                0                0    0
## 103               0                0                0                0    0
## 104               0                0                0                0    0
## 105               0                0                0                0    0
## 106               0                1                0                0    1
## 107               0                0                0                0    0
## 108               0                0                0                1    1
## 109               0                0                0                0    0
## 110               0                0                0                0    0
## 111               0                0                0                0    0
## 112               0                0                0                0    0
## 113               1                1                1                1    4
##     class
## 1       1
## 2       1
## 3       1
## 4       1
## 5       0
## 6       1
```

```
## 7       1
## 8       1
## 9       1
## 10      0
## 11      0
## 12      0
## 13      0
## 14      1
## 15      1
## 16      1
## 17      0
## 18      1
## 19      1
## 20      1
## 21      1
## 22      1
## 23      1
## 24      1
## 25      1
## 26      0
## 27      0
## 28      0
## 29      1
## 30      0
## 31      0
## 32      1
## 33      1
## 34      0
## 35      1
## 36      1
## 37      0
## 38      1
## 39      1
## 40      0
## 41      0
## 42      0
## 43      1
## 44      1
## 45      0
## 46      0
## 47      1
## 48      1
## 49      1
## 50      0
## 51      0
## 52      0
## 53      1
## 54      0
## 55      0
## 56      1
## 57      1
## 58      0
## 59      0
## 60      0
```

```
## 61      0
## 62      0
## 63      0
## 64      0
## 65      1
## 66      0
## 67      1
## 68      0
## 69      0
## 70      0
## 71      1
## 72      0
## 73      0
## 74      1
## 75      0
## 76      0
## 77      0
## 78      1
## 79      0
## 80      0
## 81      0
## 82      0
## 83      1
## 84      0
## 85      1
## 86      0
## 87      0
## 88      0
## 89      1
## 90      0
## 91      1
## 92      0
## 93      0
## 94      1
## 95      0
## 96      0
## 97      0
## 98      0
## 99      0
## 100     0
## 101     1
## 102     0
## 103     0
## 104     0
## 105     0
## 106     0
## 107     0
## 108     0
## 109     0
## 110     0
## 111     0
## 112     0
## 113     1
```