Classification Methods for Breast Cancer Data

Noah Jones

10/12/2021

The goal of this project was to compare different classification techniques, namely variations of Logistic Regression and KNN, in terms of effectiveness in predicting whether a patient's breast cancer diagnosis is Malignant or Benign (coded as M and B, respectively) based on data that includes 10 quantitative features of tumors such as radius, perimeter, and area. Data is split into testing and training sets, and misclassification rate of the testing data set is used as to evaluate model effectiveness.

```
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(gridExtra)
library(class)
library(car)
```

```
## Loading required package: carData
```

Part 1

Basic summary statistics.

```
breastcancer <- read.csv("BreastCancer.csv")
dim(breastcancer)
```

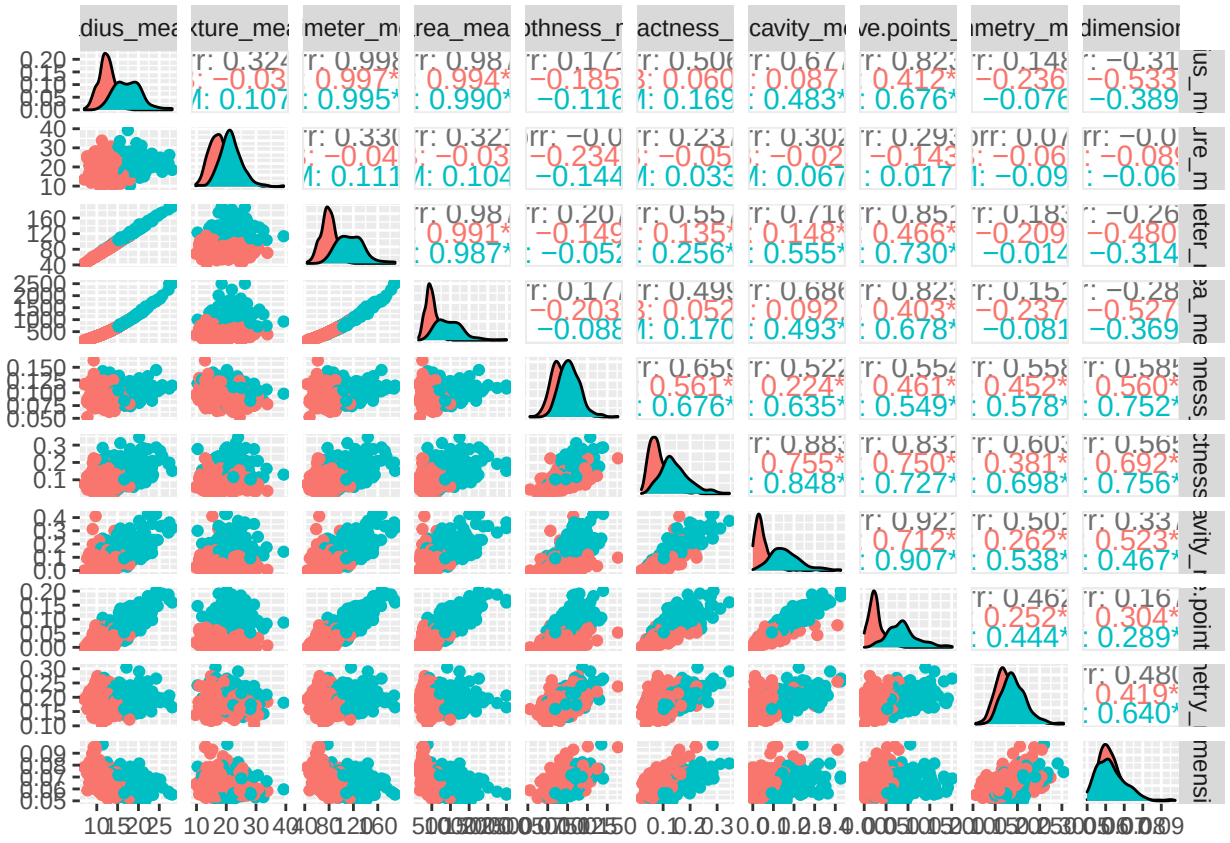
```
## [1] 569  12
```

```
summary(breastcancer)
```

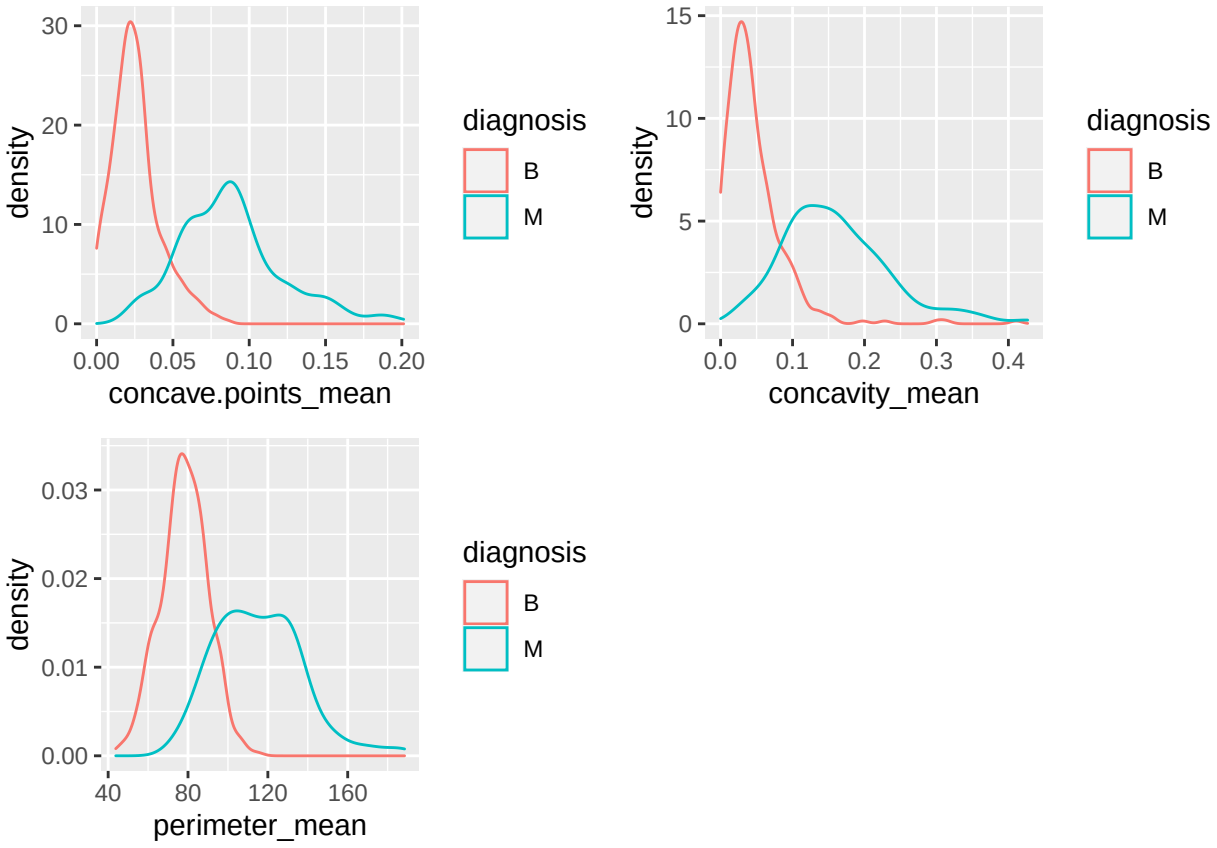
```
##        X       diagnosis  radius_mean      texture_mean   perimeter_mean
##  Min.   :  1   B:357     Min.   : 6.981   Min.   : 9.71   Min.   : 43.79
##  1st Qu.:143   M:212     1st Qu.:11.700   1st Qu.:16.17   1st Qu.: 75.17
##  Median :285             Median :13.370   Median :18.84   Median : 86.24
##  Mean   :285             Mean   :14.127   Mean   :19.29   Mean   : 91.97
##  3rd Qu.:427             3rd Qu.:15.780   3rd Qu.:21.80   3rd Qu.:104.10
##  Max.   :569             Max.   :28.110   Max.   :39.28   Max.   :188.50
##    area_mean      smoothness_mean   compactness_mean  concavity_mean
##  Min.   : 143.5   Min.   :0.05263   Min.   :0.01938   Min.   :0.00000
##  1st Qu.: 420.3   1st Qu.:0.08637   1st Qu.:0.06492   1st Qu.:0.02956
##  Median : 551.1   Median :0.09587   Median :0.09263   Median :0.06154
##  Mean   : 654.9   Mean   :0.09636   Mean   :0.10434   Mean   :0.08880
##  3rd Qu.: 782.7   3rd Qu.:0.10530   3rd Qu.:0.13040   3rd Qu.:0.13070
##  Max.   :2501.0   Max.   :0.16340   Max.   :0.34540   Max.   :0.42680
##  concave.points_mean symmetry_mean    fractal_dimension_mean
##  Min.   :0.00000     Min.   :0.1060   Min.   :0.04996
##  1st Qu.:0.02031     1st Qu.:0.1619   1st Qu.:0.05770
##  Median :0.03350     Median :0.1792   Median :0.06154
##  Mean   :0.04892     Mean   :0.1812   Mean   :0.06280
##  3rd Qu.:0.07400     3rd Qu.:0.1957   3rd Qu.:0.06612
##  Max.   :0.20120     Max.   :0.3040   Max.   :0.09744
```

Graphically identifying three most significant predictors for a patient's diagnosis - i.e., for which predictors is there the least overlap between the Malignant and Benign categories.

```
ggpairs(data = breastcancer[,c(-1,-2)], aes(color = breastcancer$diagnosis))
```



```
g1 <- ggplot(breastcancer, aes(concave.points_mean, color = diagnosis)) + geom_density()
g2 <- ggplot(breastcancer, aes(concavity_mean, color = diagnosis)) + geom_density()
g3 <- ggplot(breastcancer, aes(perimeter_mean, color = diagnosis)) + geom_density()
grid.arrange(g1,g2,g3, nrow=2)
```



Splitting data into testing and training sets, and running K-nearest-neighbor (KNN) classification for k = 1,3,5,7,9, and 11.

```
set.seed(1128)
train_indices <- sample(1:nrow(breastcancer), 400, replace = F)
test_indices <- 1:nrow(breastcancer)
test_indices <- test_indices[-train_indices]
predictors <- breastcancer[,c(5,9,10)]
train_predictors <- predictors[train_indices,]
test_predictors <- predictors[test_indices,]
train_outcomes <- breastcancer$diagnosis[train_indices]
test_outcomes <- breastcancer$diagnosis[test_indices]
knn_unscaled_results <- vector(mode = "list", length = 6)
knn_k_values <- c(1,3,5,7,9,11)
for (i in 1:length(knn_k_values)) {
  knn_unscaled_results[i] <- list(knn(train_predictors,test_predictors,train_outcomes,k = knn_k_values[
}
```

Reporting misclassification rate for the 6 KNN models.

```
for(i in 1:length(knn_k_values)){
  cat("Misclassification Rate for k = ",knn_k_values[i],": ",mean(knn_unscaled_results[[i]] != test_out
}
```

```
## Misclassification Rate for k = 1: 0.1715976
## Misclassification Rate for k = 3: 0.1420118
## Misclassification Rate for k = 5: 0.1301775
## Misclassification Rate for k = 7: 0.1360947
## Misclassification Rate for k = 9: 0.1420118
## Misclassification Rate for k = 11: 0.1420118
```

So our best k is 5, which has the lowest misclassification rate.

Repeating the above analysis after scaling the predictors.

```
scaled_predictors <- scale(predictors)
scaled_train_predictors <- scaled_predictors[train_indices,]
scaled_test_predictors <- scaled_predictors[test_indices,]
knn_scaled_results <- vector(mode = "list", length = 6)
for (i in 1:length(knn_k_values)) {
  knn_scaled_results[i] <-
    list(knn(scaled_train_predictors,scaled_test_predictors,train_outcomes,k = knn_k_values[i]))
}
```

Reporting misclassification rates for scaled predictors.

```
for(i in 1:length(knn_k_values)){
  cat("Misclassification Rate for k = ",knn_k_values[i],": ",
      mean(knn_scaled_results[[i]] != test_outcomes), "\n", sep = '')
}
```

```
## Misclassification Rate for k = 1: 0.1242604
## Misclassification Rate for k = 3: 0.1005917
## Misclassification Rate for k = 5: 0.1065089
## Misclassification Rate for k = 7: 0.1005917
## Misclassification Rate for k = 9: 0.1005917
## Misclassification Rate for k = 11: 0.09467456
```

With scaled predictors, we note that all of our misclassification rates are lower than our "best" k from part d., and also that k = 11 is our best predictor in the scaled case.

Repeating the above analysis, but including all predictors instead of just the 3 we identified as being "most significant."

```
all_predictors <- breastcancer[,c(-1,-2)]
all_train_predictors <- all_predictors[train_indices,]
all_test_predictors <- all_predictors[test_indices,]
knn_unscaled_results_all_predictors <- vector(mode = "list", length = 6)
for (i in 1:length(knn_k_values)) {
  knn_unscaled_results_all_predictors[i] <- list(knn(all_train_predictors,all_test_predictors,train_out
}
for(i in 1:length(knn_k_values)){
  cat("Misclassification Rate for all predictors, unscaled, for k = ",knn_k_values[i],": ",mean(knn_uns
}
```

```
## Misclassification Rate for all predictors, unscaled, for k = 1: 0.1775148
## Misclassification Rate for all predictors, unscaled, for k = 3: 0.1538462
## Misclassification Rate for all predictors, unscaled, for k = 5: 0.1538462
## Misclassification Rate for all predictors, unscaled, for k = 7: 0.147929
## Misclassification Rate for all predictors, unscaled, for k = 9: 0.1538462
## Misclassification Rate for all predictors, unscaled, for k = 11: 0.1538462
```

```
all_scaled_predictors <- scale(all_predictors)
all_scaled_train_predictors <- all_scaled_predictors[train_indices,]
all_scaled_test_predictors <- all_scaled_predictors[test_indices,]
knn_scaled_results_all_predictors <- vector(mode = "list", length = 6)
for (i in 1:length(knn_k_values)) {
  knn_scaled_results_all_predictors[i] <- list(knn(all_scaled_train_predictors,all_scaled_test_predicto
}
for(i in 1:length(knn_k_values)){
  cat("Misclassification Rate for all predictors, scaled, for k = ",knn_k_values[i],": ",mean(knn_scale
}
```

```
## Misclassification Rate for all predictors, scaled, for k = 1: 0.1242604
## Misclassification Rate for all predictors, scaled, for k = 3: 0.0591716
## Misclassification Rate for all predictors, scaled, for k = 5: 0.0591716
## Misclassification Rate for all predictors, scaled, for k = 7: 0.05325444
## Misclassification Rate for all predictors, scaled, for k = 9: 0.0591716
## Misclassification Rate for all predictors, scaled, for k = 11: 0.0591716
```

In the unscaled case, our "best k" is 7, however the KNN model with all predictors has a higher misclassification rate than the model with only the 3 significant predictors for all k. In the scaled case, our "best k" is also 7, and the KNN model with all predictors has a lower misclassification rate than the model with only the 3 significant predictors for all k except for k = 1, suggesting that using all predictors might yield better results.

Modeling via Logistic Regression

Running logestic regression model for all numerical predictors, followed by a reporting of confusion matrices and misclassification rates for training and testing data sets.

```r
m1 <- glm(diagnosis~radius_mean+texture_mean+perimeter_mean+area_mean+smoothness_mean
          +compactness_mean+concavity_mean+concave.points_mean+symmetry_mean
          +fractal_dimension_mean,data = breastcancer[train_indices,],
          family = "binomial")
```

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

```r
summary(m1)
```

```
##
## Call:
## glm(formula = diagnosis ~ radius_mean + texture_mean + perimeter_mean +
##     area_mean + smoothness_mean + compactness_mean + concavity_mean +
##     concave.points_mean + symmetry_mean + fractal_dimension_mean,
##     family = "binomial", data = breastcancer[train_indices, ])
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.98124  -0.12202  -0.02629   0.00094   2.76344
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)                2.68098   18.58003   0.144   0.8853
## radius_mean               -4.21289    5.15561  -0.817   0.4138
## texture_mean               0.39058    0.08706   4.486 7.25e-06 ***
## perimeter_mean             0.05024    0.72192   0.070   0.9445
## area_mean                  0.05881    0.02603   2.259   0.0239 *
## smoothness_mean          102.18212   43.94559   2.325   0.0201 *
## compactness_mean          12.06131   27.91246   0.432   0.6657
## concavity_mean            17.34440   11.36669   1.526   0.1270
## concave.points_mean       29.42421   38.02850   0.774   0.4391
## symmetry_mean             19.66157   15.66165   1.255   0.2093
## fractal_dimension_mean -165.06168  113.63982  -1.452   0.1464
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 530.26  on 399  degrees of freedom
## Residual deviance:  89.82  on 389  degrees of freedom
## AIC: 111.82
##
## Number of Fisher Scoring iterations: 9
```

```r
train_logit_probabilities <- predict(m1,type="response")
test_logit_probabilities <- predict(m1,newdata = breastcancer[test_indices,],type="response")
train_logit_results <- ifelse(train_logit_probabilities > 0.5, "M", "B")
test_logit_results <- ifelse(test_logit_probabilities > 0.5, "M", "B")
```

Training Data Confusion Matrix:

```r
table(train_outcomes,train_logit_results)
```

```
##               train_logit_results
## train_outcomes   B   M
##              B 241   8
##              M   9 142
```

Testing Data Confusion Matrix:

```r
table(test_outcomes,test_logit_results)
```

```
##              test_logit_results
## test_outcomes   B   M
##             B 103   5
##             M   8  53
```

Misclassification Rates:

```r
mean(train_outcomes!=train_logit_results)
```

```
## [1] 0.0425
```

```r
mean(test_outcomes!=test_logit_results)
```

```
## [1] 0.07692308
```

Repeating the above analysis after scaling the predictors.

```r
scaled_breastcancer <- data.frame(breastcancer[,c(1,2)],scale(breastcancer[,c(-1,-2)]))
m2 <- glm(diagnosis~radius_mean+texture_mean+perimeter_mean+area_mean+smoothness_mean
          +compactness_mean+concavity_mean+concave.points_mean+symmetry_mean
          +fractal_dimension_mean,data = scaled_breastcancer[train_indices,],
          family = "binomial")
```

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

```r
summary(m2)
```

```
##
## Call:
## glm(formula = diagnosis ~ radius_mean + texture_mean + perimeter_mean +
##     area_mean + smoothness_mean + compactness_mean + concavity_mean +
##     concave.points_mean + symmetry_mean + fractal_dimension_mean,
##     family = "binomial", data = scaled_breastcancer[train_indices,
##         ])
##
## Deviance Residuals:
```

```
##      Min        1Q    Median        3Q       Max
## -1.98124  -0.12202  -0.02629   0.00094   2.76344
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)              1.1119     0.8934   1.245   0.2133
## radius_mean            -14.8464    18.1686  -0.817   0.4138
## texture_mean             1.6799     0.3745   4.486 7.25e-06 ***
## perimeter_mean           1.2207    17.5418   0.070   0.9445
## area_mean               20.6951     9.1608   2.259   0.0239 *
## smoothness_mean          1.4371     0.6181   2.325   0.0201 *
## compactness_mean         0.6370     1.4741   0.432   0.6657
## concavity_mean           1.3827     0.9062   1.526   0.1270
## concave.points_mean      1.1417     1.4756   0.774   0.4391
## symmetry_mean            0.5390     0.4294   1.255   0.2093
## fractal_dimension_mean  -1.1654     0.8023  -1.452   0.1464
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 530.26  on 399  degrees of freedom
## Residual deviance:  89.82  on 389  degrees of freedom
## AIC: 111.82
##
## Number of Fisher Scoring iterations: 9
```

```
train_scaled_logit_probabilities <- predict(m2,type="response")
test_scaled_logit_probabilities <- predict(m2,newdata = scaled_breastcancer[test_indices,],type="respons
train_scaled_logit_results <- ifelse(train_scaled_logit_probabilities > 0.5, "M", "B")
test_scaled_logit_results <- ifelse(test_scaled_logit_probabilities > 0.5, "M", "B")
```

Reporting confusion matrices and misclassification rates for scaled data.

Scaled Training Data Confusion Matrix:

```
table(train_outcomes,train_scaled_logit_results)
```

```
##                train_scaled_logit_results
## train_outcomes   B   M
##              B 241   8
##              M   9 142
```

Scaled Testing Data Confusion Matrix:

```
table(test_outcomes,test_scaled_logit_results)
```

```
##               test_scaled_logit_results
## test_outcomes   B   M
##             B 103   5
##             M   8  53
```

Misclassification Rates:

```
mean(train_outcomes!=train_scaled_logit_results)
```

```
## [1] 0.0425
```

```
mean(test_outcomes!=test_scaled_logit_results)
```

```
## [1] 0.07692308
```

Comparing success rates of different models and acknowledging shortcomings of the project, particularly in the multicollinearity of our predictors.

```
mean(test_logit_probabilities==test_scaled_logit_probabilities)
```

```
## [1] 0.2485207
```

```
mean(test_logit_results==test_scaled_logit_results)
```

```
## [1] 1
```

We note that our scaled and unscaled logistic models produced different probabilities, but still yielded the same results with the 0.5 cutoff (see code above).

If we compare the results of the KNN and Logistic regression models off of misclassification rate alone, then the scaled KNN model with all predictors and k = 7 would be our best model, given its misclassification rate of 0.05325444. I am hesitant to say that this is our best possible model, because I believe there is some level of multicollinearity involved with the models which use all predictors. We can see this in our logistic regression models, in which only 3 of our predictors have statistical significance, and also in the VIF function, which shows us some pretty alarmingly large correlation between some of our predictor variables (see code below).

```
vif(m1)
```

```
##            radius_mean           texture_mean         perimeter_mean
##              991.655019                1.937031              799.268740
##              area_mean        smoothness_mean       compactness_mean
##              169.273768                5.798963               16.242400
##         concavity_mean    concave.points_mean          symmetry_mean
##                6.027088                6.479460                2.389118
## fractal_dimension_mean
##               11.563373
```