SDS 385 Report

Akshay Kumar Varanasi (av32826)

Report

Description of the data

Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image and the 3-dimensional space is that described in: [K. P. Bennett and O. L. Mangasarian: “Robust Linear Programming Discrimination of Two Linearly Inseparable Sets”, Optimization Methods and Software 1, 1992, 23-34].

This database is also available through the UW CS ftp server: ftp ftp.cs.wisc.edu cd math-prog/cpo-dataset/machine-learn/WDBC/

Also can be found on UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29 (https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29)

Attribute Information:

1. ID number 2) Diagnosis (M = malignant, B = benign) 3-32)

Ten real-valued features are computed for each cell nucleus:

a. radius (mean of distances from center to points on the perimeter) b) texture (standard deviation of gray-scale values) c) perimeter d) area e) smoothness (local variation in radius lengths) f) compactness (perimeter^2 / area - 1.0) g) concavity (severity of concave portions of the contour) h) concave points (number of concave portions of the contour) i) symmetry j) fractal dimension (“coastline approximation” - 1)

The mean, standard error and “worst” or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius.

All feature values are recoded with four significant digits.

Missing attribute values: none

Class distribution: 357 benign, 212 malignant

Question

Fit a logistic regression model to predict the diagnosis using the different features computed for each cell nucleus. In the process of doing that, we get insight about the data for example what features are important and what are not for the diagnosis.

Answer

Call the necessary the libraries

```
library(ggplot2)
library(plotROC)
```

Before we start the analysis we first do little preprocessing like removing NAs and consider only the columns with “mean” values for simplicity.

```
# Reading the data from csv file
data = read.csv("breastcancer_data.csv")

# Removing NA values if any and removing last column which contains NAs
data_cleaned = na.omit(data[c(-length(data))])

# considering only mean values for the data analysis
cancer_data = data_cleaned[,1:12]
head(cancer_data)
```

```
##         id diagnosis radius_mean texture_mean perimeter_mean area_mean
## 1   842302         M       17.99        10.38         122.80    1001.0
## 2   842517         M       20.57        17.77         132.90    1326.0
## 3 84300903         M       19.69        21.25         130.00    1203.0
## 4 84348301         M       11.42        20.38          77.58     386.1
## 5 84358402         M       20.29        14.34         135.10    1297.0
## 6   843786         M       12.45        15.70          82.57     477.1
##   smoothness_mean compactness_mean concavity_mean concave.points_mean
## 1         0.11840          0.27760         0.3001             0.14710
## 2         0.08474          0.07864         0.0869             0.07017
## 3         0.10960          0.15990         0.1974             0.12790
## 4         0.14250          0.28390         0.2414             0.10520
## 5         0.10030          0.13280         0.1980             0.10430
## 6         0.12780          0.17000         0.1578             0.08089
##   symmetry_mean fractal_dimension_mean
## 1        0.2419                0.07871
## 2        0.1812                0.05667
## 3        0.2069                0.05999
## 4        0.2597                0.09744
## 5        0.1809                0.05883
## 6        0.2087                0.07613
```

Before we fit the model, we divide the data into training and testing data.

```
train_fraction <- 0.5 # fraction of data for training purposes
set.seed(126) # set the seed to make the partition reproductible
train_size <- floor(train_fraction * nrow(cancer_data)) # number of observation
s in training set

train_indices <- sample(1:nrow(cancer_data), size = train_size)
train_data <- cancer_data[train_indices, ] # get training data
test_data <- cancer_data[-train_indices, ] # get test data
```

Now we fit the model with using all the features initially and then we remove one by one gradually till we get a reasonable model

```
glm_out <- glm(
diagnosis ~ radius_mean + texture_mean + perimeter_mean + area_mean + smoothnes
s_mean + compactness_mean + concavity_mean + concave.points_mean + symmetry_mea
n + fractal_dimension_mean,
data = train_data,
family = binomial
) # family = binomial required for logistic regression
```

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

```
summary(glm_out)
```

```
##
## 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 = train_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.67271  -0.13664  -0.04140   0.00738   2.88986
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -8.2796    17.0123  -0.487   0.6265
## radius_mean              3.1052     4.3454   0.715   0.4749
## texture_mean             0.4159     0.1002   4.153 3.28e-05 ***
## perimeter_mean          -0.8485     0.6119  -1.387   0.1656
## area_mean                0.0378     0.0229   1.651   0.0988 .
## smoothness_mean         17.8194    46.8091   0.381   0.7034
## compactness_mean        24.9075    26.4741   0.941   0.3468
## concavity_mean          -0.3882    11.4035  -0.034   0.9728
## concave.points_mean     88.9512    45.0613   1.974   0.0484 *
## symmetry_mean           16.2933    17.8799   0.911   0.3622
## fractal_dimension_mean -25.5453   117.0377  -0.218   0.8272
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 382.593  on 283  degrees of freedom
## Residual deviance:  76.155  on 273  degrees of freedom
## AIC: 98.155
##
## Number of Fisher Scoring iterations: 9
```

After we successively remove predictors until only predictors with a p value less than 0.1 remain, we get the following model.

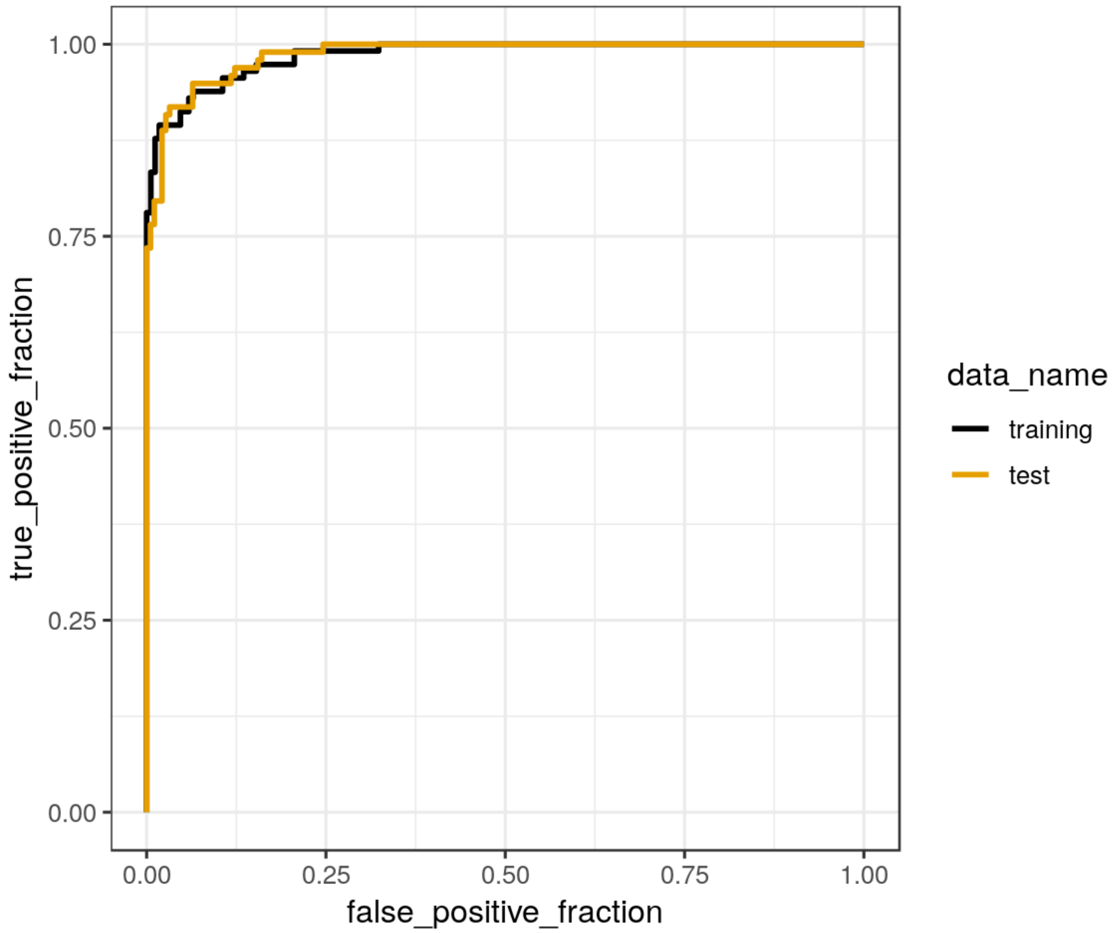
```
# diagnosis ~ texture_mean + area_mean + concave.points_mean

glm_out <- glm(
diagnosis ~ texture_mean + area_mean + concave.points_mean,
data = train_data,
family = binomial
) # family = binomial required for logistic regression
summary(glm_out)
```

```
##
## Call:
## glm(formula = diagnosis ~ texture_mean + area_mean + concave.points_mean,
##     family = binomial, data = train_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.20915  -0.15609  -0.05011   0.03123   2.74634
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -16.774583   2.669581  -6.284 3.31e-10 ***
## texture_mean          0.365861   0.085759   4.266 1.99e-05 ***
## area_mean             0.006866   0.001948   3.525 0.000424 ***
## concave.points_mean  97.582938  18.451306   5.289 1.23e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 382.593  on 283  degrees of freedom
## Residual deviance:  82.797  on 280  degrees of freedom
## AIC: 90.797
##
## Number of Fisher Scoring iterations: 8
```

We create two different data frames one for training and one for test data. Now we see how good is this model on data by plotting the ROC curve.

```
# results data frame for training data
df_train <- data.frame(
predictor = predict(glm_out, train_data),
known_truth = train_data$diagnosis,
data_name = "training"
)
# results data frame for test data
df_test <- data.frame(
predictor = predict(glm_out, test_data),
known_truth = test_data$diagnosis,
data_name = "test"
)
df_combined <- rbind(df_train, df_test)
ggplot(df_combined, aes(d = known_truth, m = predictor, color = data_name)) +
geom_roc(n.cuts = 0) +
scale_color_colorblind()
```

```
## Warning in verify_d(data$d): D not labeled 0/1, assuming B = 0 and M = 1!
```



After looking at ROC curves we see that the model fits and predicts quite well on the data. Even the AUC values say the same thing that model really good on both the training and test data.

```
p <- ggplot(df_combined, aes(d = known_truth, m = predictor, color = data_name)
) +
geom_roc(n.cuts = 0)
data_name <- unique(df_combined$data_name)
data_info <- data.frame(
data_name,
group = order(data_name)
)
left_join(data_info, calc_auc(p)) %>%
select(-group, -PANEL) %>%
arrange(desc(AUC))
```

```
## Warning in verify_d(data$d): D not labeled 0/1, assuming B = 0 and M = 1!
## Joining, by = "group"
```

```
##   data_name       AUC
## 1      test 0.9865219
## 2  training 0.9856037
```

So by fitting the model and testing it we see that the model performs really good. This is because the diagnosis is dependent on the following three mean values: texture of the nuclei image , area of the nuclei , concave points. This is true because cancer is due to irregular nuclei size like large nuclei(area), texture is different as it is not of same color uniformly and the shape is not circular so has many concave points.