Project 2: Breast Cancer Diagnosis

Xinran Sun

3/17/2022

Objectives

A mammogram is an X-ray image of breast tissue. It can help save lives because it is easier to treat breast cancer in its early stages before the cancer is big enough to detect or cause symptoms. However, a wrong diagnosis can have a negative impact on patients. For example, if there is a false-positive test result, the doctor sees something that looks like cancer but is not. This could result in overtreatment that causes unnecessary side effects on patients. On the other hand, false-negative test result occurs when a doctor misses cancer tissues, which may delay the treatment. Therefore, building a model that gives an accurate classification of the tissue images is necessary to give proper treatment. In our study, we collected 569 images from both malignant and benign cancer tissues. Our goal is to build a predictive model to facilitate cancer diagnosis.

Dataset

Our data set consists of 569 rows, with 357 benign and 212 malignant. We denote 0 for benign and 1 for malignant. We also have 30 columns representing the features of the tissue images. They include the mean, standard deviation, and the largest values of the distributions of the following 10 features computed for the cell nuclei:

- radius (mean of distances from center to points on the perimeter)
- texture (standard deviation of gray-scale values)
- perimeter
- area
- smoothness (local variation in radius lengths)
- compactness ($perimeter^2/area$ - 1.0)
- concavity (severity of concave portions of the contour)
- concave points (number of concave portions of the contour)
- symmetry
- fractal dimension ("coastline approximation" - 1)

EDA

Before building the model, we want have a close look at the dataset. Therefore, we first examine the correlation between variables. The squares with dark color in the correlation plot has strong correlation with each other. We can see there is very strong correlation between radius, perimeter and area across mean, standard deviation, and the largest values. We decided to drop some variables with correlation larger than 0.7. The variables we dropped are perimeter_mean, area_mean, compactness_mean, concave_points_mean, perimeter_se, area_se, radius_worst, texture_worst, perimeter_worst, area_worst, and concavity_worst (11 variables).

After that, we built feature plot to analyze the relationship between variables after removing the variables with high correlation. From this plot, we can see that there are no strong relationship between variables after removing. We also found that the points for benign tissues are often locate at left-bottom side, which indicates the benign tissues usually have smaller feature values compared to malignant tissues.

We also calculated the mean of each variables to compare values between benign and malignant cases. According to the average values of the mean of each feature, we can find that benign tissues have smaller values compared to malignant tissues, except for fractal dimension. There is no general pattern for the average values of the standard deviations. Based the average values of the largest value of each feature, we can find that benign tissues have smaller largest values compared to malignant tissues.

Methods

Logistic Model

Let y be the vector with 569 binary response variable, X be the 569×30 matrix with 30 numerical explanatory variables, and β be the vector with 30 corresponding coefficients. We also have β_0 as the intercepts.

For our logistic model, the probability of ith row be a malignant tissue is given by:

$$P(y_i = 1|X_i) = \frac{e^{\beta_0+\beta X_i}}{1 + e^{\beta_0+\beta X_i}}.$$

For likelihood function is:

$$L(\beta_0, \beta) = \prod_{i=1}^{n} [(\frac{e^{\beta_0+\beta X_i}}{1 + e^{\beta_0+\beta X_i}})^{y_i} (\frac{1}{1 + e^{\beta_0+\beta X_i}})^{1-y_i}].$$

Maximizing the likelihood is equivalent to maximizing the log likelihood:

$$f(\beta_0, \beta) = \sum_{i=1}^{n} [y_i(\beta_0 + \beta X_i) - \log(1 + e^{\beta_0+\beta X_i})].$$

The gradient of this function is:

$$\nabla f(\beta_0, \beta) = \begin{pmatrix} \sum_{i=1}^{n} y_i - p_i \\ \sum_{i=1}^{n} X_1(y_i - p_i) \\ ... \\ \sum_{i=1}^{n} X_n(y_i - p_i) \end{pmatrix} = X^T(y_i - p_i)$$

where $p_i = P(y_i = 1|X_i)$ as mentioned in previous probability function.

The Hessian is given by

$$\nabla^2 f(\beta_0, \beta) = -X^T W X$$

where $W = p_i(1 - p_i)$.

Newton-Raphson Algorithm

Path-wise Coordinate-wise Optimization Algorithm

To obtain a path of solutions with a descending sequence of λ's in a logistic-LASSO model, we can implement a path-wise coordinate-wise optimization algorithm which contains the following steps:

- Step 1: Find the smallest value λ for which all the estimated β are 0, defined as λ_{max}.
- Step 2: Define a fine sequence $\lambda_{max} \geq \lambda_1 \geq ...\lambda_{min} \geq 0$.
- Step 3: To estimate coefficients for the current λ_{k+1}, implement coordinate descent algorithm using the computed coefficients of the previous λ_k (warm start) as coefficient start values.

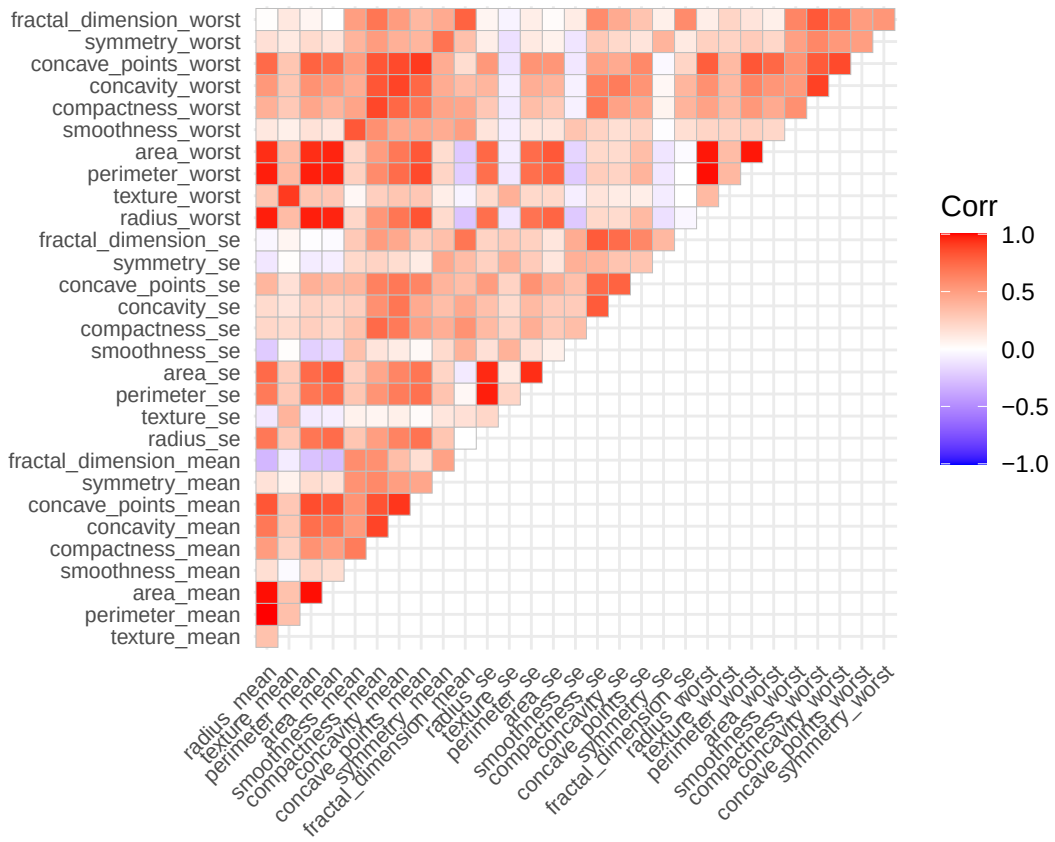
Results

Conclusions

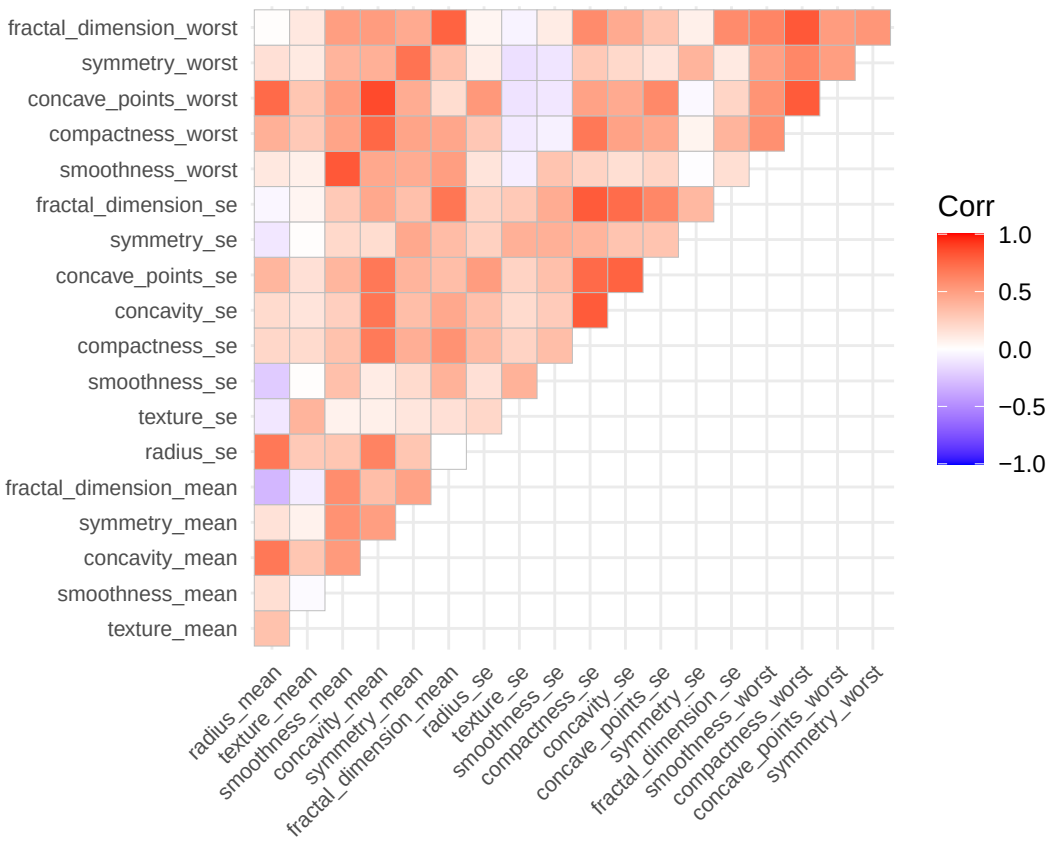
Appendix

data import and data clean

```
#load the data
breast = read.csv("breast-cancer.csv") %>%
  janitor::clean_names() %>%
  dplyr::select(-1, -33) %>% #drop id and NA columns
  mutate(diagnosis = recode(diagnosis, "M" = 1, "B" = 0))
#check collinearity
corr = breast[2:31] %>%
  cor()
ggcorrplot(corr, type = "upper", tl.cex = 8)
```




```
#remove some highly correlated variables
breast_dat <- breast %>% dplyr::select(-area_se, -perimeter_se, -area_worst, -perimeter_mean, -perimeter
corr1 = breast_dat[2:20] %>%
  cor()
ggcorrplot(corr1, type = "upper", tl.cex = 8)
```




```
#partition data into training and test data
trainRows <- createDataPartition(y = breast_dat$diagnosis, p = 0.8, list = FALSE)
breast_train <- breast_dat[trainRows, ]
breast_test <-  breast_dat[-trainRows, ]
head(breast_dat, 5)
```

```
##   diagnosis radius_mean texture_mean smoothness_mean concavity_mean
## 1         1       17.99        10.38         0.11840         0.3001
## 2         1       20.57        17.77         0.08474         0.0869
## 3         1       19.69        21.25         0.10960         0.1974
## 4         1       11.42        20.38         0.14250         0.2414
## 5         1       20.29        14.34         0.10030         0.1980
##   symmetry_mean fractal_dimension_mean radius_se texture_se smoothness_se
## 1        0.2419                0.07871    1.0950     0.9053      0.006399
## 2        0.1812                0.05667    0.5435     0.7339      0.005225
## 3        0.2069                0.05999    0.7456     0.7869      0.006150
## 4        0.2597                0.09744    0.4956     1.1560      0.009110
## 5        0.1809                0.05883    0.7572     0.7813      0.011490
##   compactness_se concavity_se concave_points_se symmetry_se
## 1        0.04904      0.05373           0.01587     0.03003
## 2        0.01308      0.01860           0.01340     0.01389
## 3        0.04006      0.03832           0.02058     0.02250
## 4        0.07458      0.05661           0.01867     0.05963
## 5        0.02461      0.05688           0.01885     0.01756
##   fractal_dimension_se smoothness_worst compactness_worst concave_points_worst
## 1             0.006193           0.1622            0.6656               0.2654
## 2             0.003532           0.1238            0.1866               0.1860
```

```
## 3             0.004571          0.1444           0.4245                0.2430
## 4             0.009208          0.2098           0.8663                0.2575
## 5             0.005115          0.1374           0.2050                0.1625
##   symmetry_worst fractal_dimension_worst
## 1         0.4601                 0.11890
## 2         0.2750                 0.08902
## 3         0.3613                 0.08758
## 4         0.6638                 0.17300
## 5         0.2364                 0.07678
```

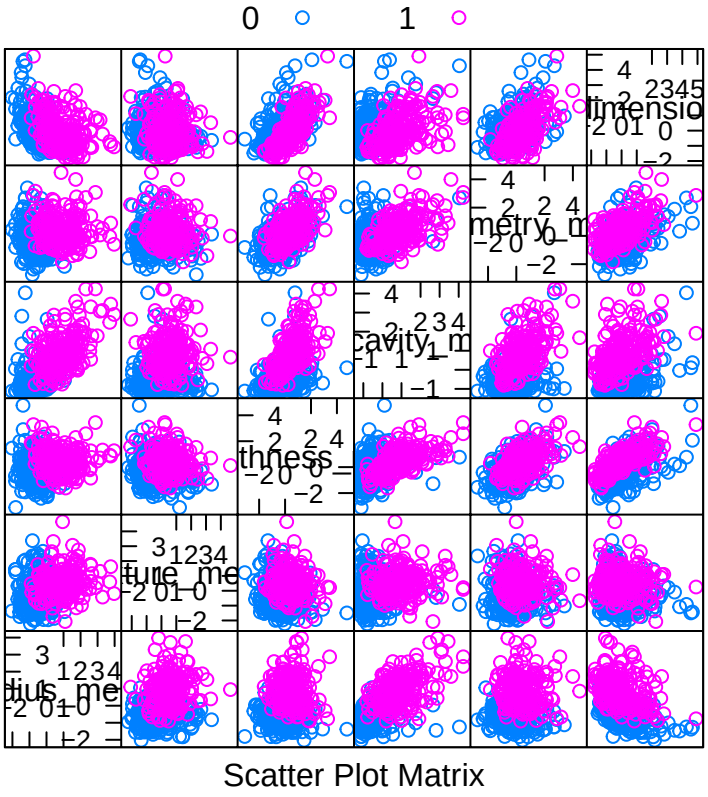
```
r = dim(breast_dat)[1] #row number
c = dim(breast_dat)[2] #column number
var_names = names(breast_dat)[-c(1,2)] #variable names

standardize = function(col) {
  mean = mean(col)
  sd = sd(col)
  return((col - mean)/sd)
}
stand_df = breast_dat %>%
  dplyr::select(radius_mean:fractal_dimension_worst) %>%
  map_df(.x = ., standardize) #standardize
X = stand_df #predictors
y = breast_dat[,1]#response
```

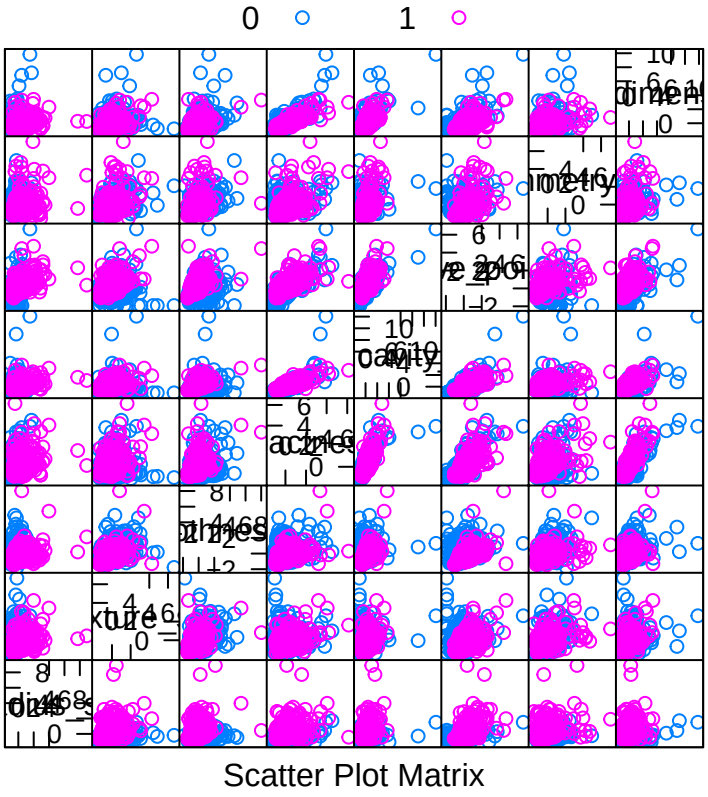
Feature plot

```
data = cbind(y,X)

featurePlot(x = data[, 2:7],
            y = factor(data$y),
            plot = "pairs",
            auto.key = list(columns = 2)
)
```

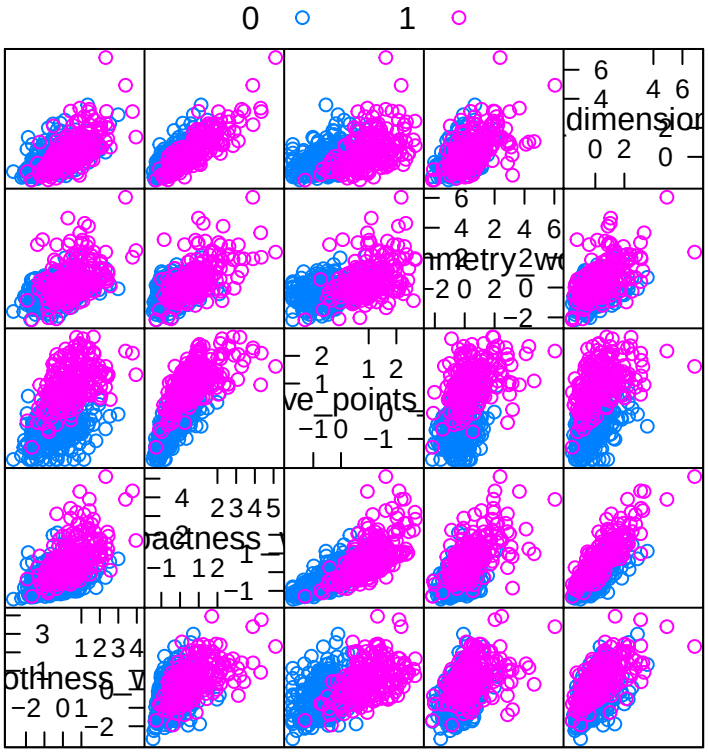



```
featurePlot(x = data[, 8:15],
            y = factor(data$y),
            plot = "pairs",
            auto.key = list(columns = 2)
)
```



Scatter Plot Matrix

```
featurePlot(x = data[, 16:20],
            y = factor(data$y),
            plot = "pairs",
            auto.key = list(columns = 2)
)
```




```
mean_data = breast_dat %>%
  group_by(diagnosis) %>%
  summarise(across(radius_mean: fractal_dimension_worst, ~ mean(.x, na.rm = TRUE)))
mean_data
```

```
## # A tibble: 2 x 20
##   diagnosis radius_mean texture_mean smoothness_mean concavity_mean
##       <dbl>       <dbl>        <dbl>           <dbl>          <dbl>
## 1         0        12.1         17.9          0.0925         0.0461
## 2         1        17.5         21.6          0.103          0.161
## # ... with 15 more variables: symmetry_mean <dbl>,
## #   fractal_dimension_mean <dbl>, radius_se <dbl>, texture_se <dbl>,
## #   smoothness_se <dbl>, compactness_se <dbl>, concavity_se <dbl>,
## #   concave_points_se <dbl>, symmetry_se <dbl>, fractal_dimension_se <dbl>,
## #   smoothness_worst <dbl>, compactness_worst <dbl>,
## #   concave_points_worst <dbl>, symmetry_worst <dbl>,
## #   fractal_dimension_worst <dbl>
```

Full logistic model

```
glm.fit <- glm(diagnosis ~ .,
               data = breast_dat,
               subset = trainRows,
               family = binomial(link = "logit"))
summary(glm.fit)
```

```
##
## Call:
## glm(formula = diagnosis ~ ., family = binomial(link = "logit"),
##     data = breast_dat, subset = trainRows)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.58838  -0.00451  -0.00006   0.00000   2.60844
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -63.1592    22.5375  -2.802  0.00507 **
## radius_mean                 0.6176     0.4871   1.268  0.20481
## texture_mean                0.6866     0.2124   3.233  0.00122 **
## smoothness_mean          -100.3132   143.6177  -0.698  0.48488
## concavity_mean            139.7323    46.9043   2.979  0.00289 **
## symmetry_mean             -25.9299    36.5783  -0.709  0.47839
## fractal_dimension_mean    -49.4241   268.9570  -0.184  0.85420
## radius_se                  38.1583    11.9873   3.183  0.00146 **
## texture_se                 -0.5054     1.3258  -0.381  0.70304
## smoothness_se               5.5676   679.6504   0.008  0.99346
## compactness_se            -23.3208   159.3643  -0.146  0.88366
## concavity_se              -67.5843    48.0578  -1.406  0.15963
## concave_points_se         202.7670   375.7727   0.540  0.58947
## symmetry_se              -349.3586   226.5825  -1.542  0.12311
## fractal_dimension_se    -3504.0553  1627.2083  -2.153  0.03129 *
## smoothness_worst           36.5632   100.9014   0.362  0.71708
## compactness_worst         -49.3256    27.1146  -1.819  0.06889 .
## concave_points_worst       82.0205    58.7687   1.396  0.16282
## symmetry_worst             61.5790    29.4867   2.088  0.03676 *
## fractal_dimension_worst   394.6559   199.9313   1.974  0.04839 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 605.35  on 455  degrees of freedom
## Residual deviance:  41.78  on 436  degrees of freedom
## AIC: 81.78
##
## Number of Fisher Scoring iterations: 11
```

```
pred <- predict(glm.fit, newdata = breast_test, type = "response")
y_test <- factor(breast_test$diagnosis)
auc_full <- auc(y_test, pred)
auc_full
```

```
## Area under the curve: 0.957
```