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 consistes 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

We built three feature plots to analyze the relationship between variables. The mean variables was plotted in first figure. From this plot, we can see that there are strong correlations between radius mean with perimeter mean and between radius mean with area mean. Perimeter mean and area mean also have strong correlation. Similarly, for the largest values, there are strong correlations between radius, perimeter and area. The correlations are reasonable because perimeter and area are calculated based on radius.

For benign cases, all of the 10 features' means and largest values are smaller than malignant cases.

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: Define a quadratic approximated objective function

$$L(\beta_0, \beta, \lambda)$$

  for λ_k using the estimated parameter at λ_{k-1} ($\lambda_{k-1} > \lambda_k$).
- Step 4: Run coordinate descendent algorithm to find the optimization defined in Step 3.

Results

Conclusions

Appendix

data import and data clean

```
#load the data
breast_dat = read.csv("breast-cancer.csv")[, -33] %>%
  janitor::clean_names() %>%
  mutate(diagnosis = recode(diagnosis, "M" = 1, "B" = 0))


head(breast_dat, 5)
```

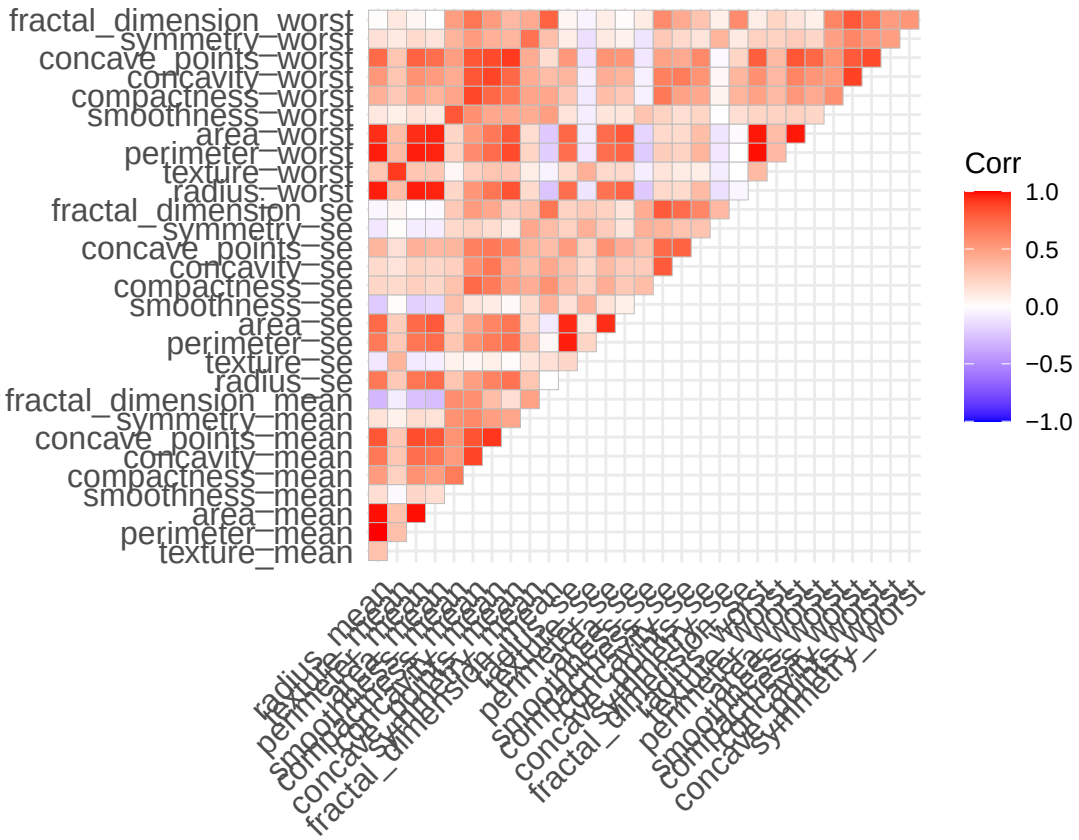
```
##         id diagnosis radius_mean texture_mean perimeter_mean area_mean
## 1   842302         1       17.99        10.38         122.80    1001.0
## 2   842517         1       20.57        17.77         132.90    1326.0
## 3 84300903         1       19.69        21.25         130.00    1203.0
## 4 84348301         1       11.42        20.38          77.58     386.1
## 5 84358402         1       20.29        14.34         135.10    1297.0
##   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
##   symmetry_mean fractal_dimension_mean radius_se texture_se perimeter_se
## 1        0.2419                0.07871    1.0950     0.9053        8.589
## 2        0.1812                0.05667    0.5435     0.7339        3.398
## 3        0.2069                0.05999    0.7456     0.7869        4.585
## 4        0.2597                0.09744    0.4956     1.1560        3.445
## 5        0.1809                0.05883    0.7572     0.7813        5.438
##   area_se smoothness_se compactness_se concavity_se concave_points_se
## 1  153.40      0.006399        0.04904      0.05373           0.01587
## 2   74.08      0.005225        0.01308      0.01860           0.01340
## 3   94.03      0.006150        0.04006      0.03832           0.02058
## 4   27.23      0.009110        0.07458      0.05661           0.01867
## 5   94.44      0.011490        0.02461      0.05688           0.01885
```

```
##   symmetry_se fractal_dimension_se radius_worst texture_worst perimeter_worst
## 1     0.03003             0.006193        25.38         17.33          184.60
## 2     0.01389             0.003532        24.99         23.41          158.80
## 3     0.02250             0.004571        23.57         25.53          152.50
## 4     0.05963             0.009208        14.91         26.50           98.87
## 5     0.01756             0.005115        22.54         16.67          152.20
##   area_worst smoothness_worst compactness_worst concavity_worst
## 1     2019.0           0.1622            0.6656          0.7119
## 2     1956.0           0.1238            0.1866          0.2416
## 3     1709.0           0.1444            0.4245          0.4504
## 4      567.7           0.2098            0.8663          0.6869
## 5     1575.0           0.1374            0.2050          0.4000
##   concave_points_worst symmetry_worst fractal_dimension_worst
## 1               0.2654         0.4601                 0.11890
## 2               0.1860         0.2750                 0.08902
## 3               0.2430         0.3613                 0.08758
## 4               0.2575         0.6638                 0.17300
## 5               0.1625         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 = as.vector(ifelse(breast_dat[,2] == "M", 1, 0))#response
```

check collinearity

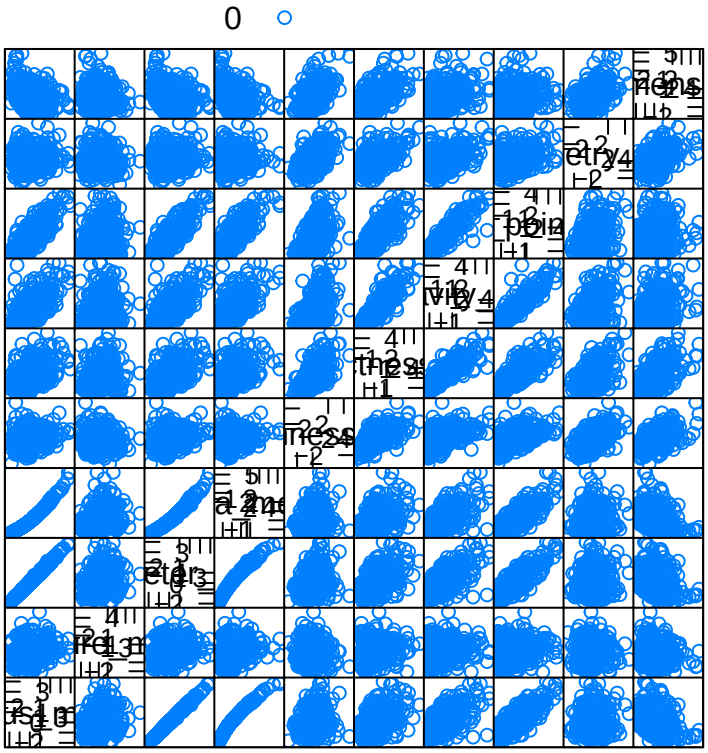
```
corr = stand_df %>%
  cor()

ggcorrplot(corr, type = "upper")
```




```
#X = stand_df %>%
  #select(radius_mean, texture_mean, smoothness_mean, compactness_mean,
         #symmetry_mean, fractal_dimension_mean, radius_se, texture_se,
         #smoothness_se, concavity_se, symmetry_se)

#corr_new = X %>% cor()
#corr_new

#ggcorrplot(corr_new, type = "upper", lab = TRUE)
```

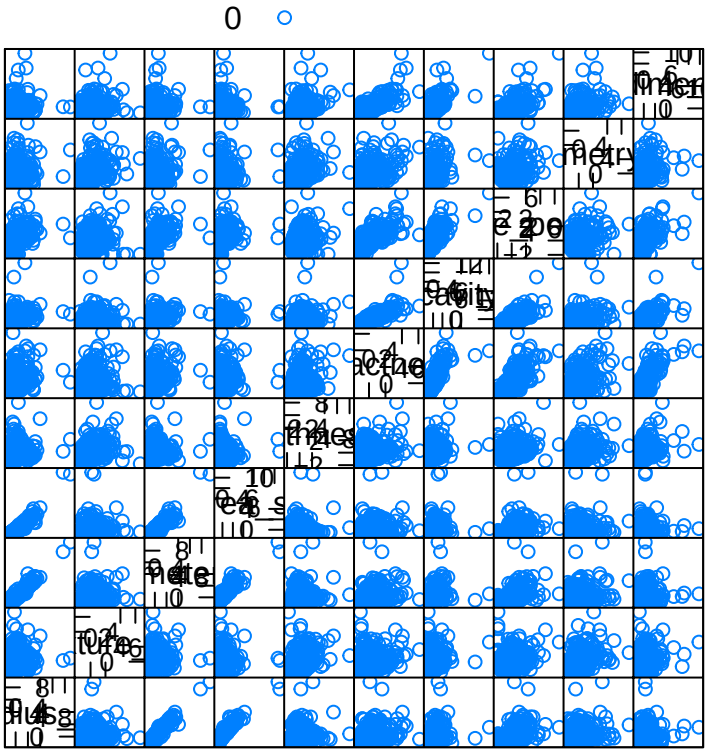
feature plot

```
data = cbind(y,X)

featurePlot(x = data[, 2:11],
            y = factor(data$y),
            plot = "pairs",
            auto.key = list(columns = 2)
)
```



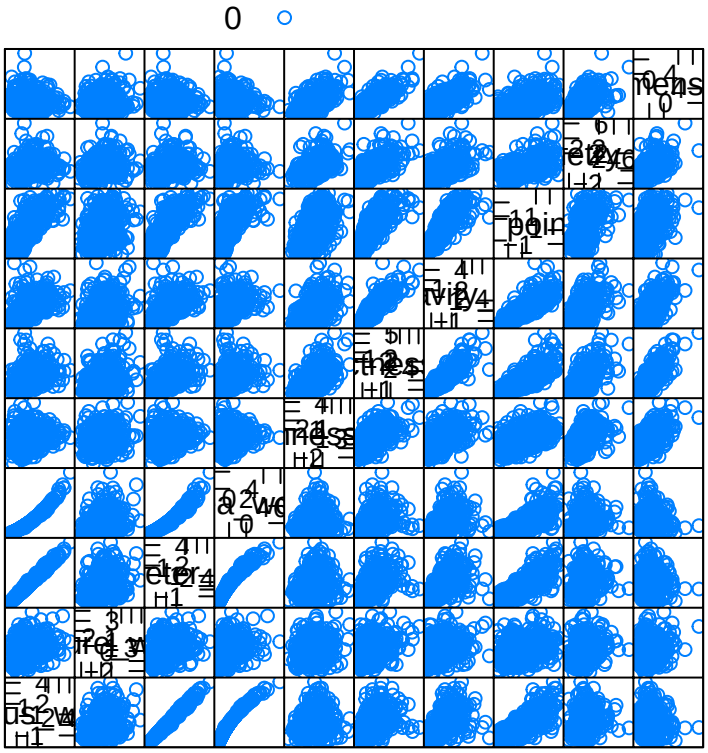
Scatter Plot Matrix

```
featurePlot(x = data[, 12:21],
            y = factor(data$y),
            plot = "pairs",
            auto.key = list(columns = 2)
)
```



Scatter Plot Matrix

```
featurePlot(x = data[, 22:31],
            y = factor(data$y),
            plot = "pairs",
            auto.key = list(columns = 2)
)
```



Scatter Plot Matrix

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

```
log_model = glm(diagnosis ~ ., family = binomial("logit"), data  = breast_dat[, -1], start = rep(0, 31))
summary(log_model)
```

```
##
## Call:
## glm(formula = diagnosis ~ ., family = binomial("logit"), data = breast_dat[,
##     -1], start = rep(0, 31))
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -3.095e-03  -2.000e-08  -2.000e-08   2.000e-08   1.538e-03
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               2.026e+02  4.842e+05   0.000    1.000
## radius_mean              -2.786e+03  5.930e+04  -0.047    0.963
## texture_mean              6.405e+01  2.686e+03   0.024    0.981
## perimeter_mean            8.056e+01  1.598e+04   0.005    0.996
## area_mean                 2.101e+01  1.262e+03   0.017    0.987
## smoothness_mean           2.098e+04  1.278e+06   0.016    0.987
## compactness_mean         -2.424e+04  1.225e+06  -0.020    0.984
## concavity_mean            1.267e+04  7.617e+05   0.017    0.987
## concave_points_mean       1.318e+04  9.767e+05   0.013    0.989
```

```
## symmetry_mean           -9.422e+03  1.976e+05  -0.048    0.962
## fractal_dimension_mean   3.011e+04  1.235e+06   0.024    0.981
## radius_se                1.396e+03  4.963e+05   0.003    0.998
## texture_se              -6.716e+01  5.532e+04  -0.001    0.999
## perimeter_se            -5.430e+02  2.837e+04  -0.019    0.985
## area_se                  4.880e+01  6.656e+03   0.007    0.994
## smoothness_se           -4.333e+04  2.757e+06  -0.016    0.987
## compactness_se           4.200e+04  3.161e+06   0.013    0.989
## concavity_se            -3.506e+04  6.996e+05  -0.050    0.960
## concave_points_se        1.373e+05  5.552e+06   0.025    0.980
## symmetry_se             -4.208e+04  4.230e+06  -0.010    0.992
## fractal_dimension_se    -3.308e+05  2.367e+07  -0.014    0.989
## radius_worst             7.389e+02  7.463e+04   0.010    0.992
## texture_worst            2.140e+01  4.569e+03   0.005    0.996
## perimeter_worst          2.939e+01  5.176e+03   0.006    0.995
## area_worst              -5.047e+00  1.115e+03  -0.005    0.996
## smoothness_worst        -2.523e+03  6.434e+05  -0.004    0.997
## compactness_worst       -3.476e+03  4.248e+05  -0.008    0.993
## concavity_worst          3.018e+03  2.901e+05   0.010    0.992
## concave_points_worst     3.069e+03  2.830e+05   0.011    0.991
## symmetry_worst           9.507e+03  3.356e+05   0.028    0.977
## fractal_dimension_worst  2.390e+04  2.637e+06   0.009    0.993
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7.5144e+02  on 568  degrees of freedom
## Residual deviance: 3.6178e-05  on 538  degrees of freedom
## AIC: 62
##
## Number of Fisher Scoring iterations: 25
```