Project BDA

Anonymous

11/14/2020

Contents

Loaded packages

```r
library(rstan)
rstan_options(auto_write = TRUE)
options(mc.cores = 3)
library(ggplot2)
library(aaltobda)
library(shinystan)
library(bayesplot)
library(loo)
library(dlookr)
library(corrplot)
library(rstanarm)
library(projpred)
library(GGally)
library(shiny)
library(gridExtra)
library(caret)
library(e1071)
library(pROC)
SEED <- 48927
```

Introduction

Breast cancer most commonly presents as a lump that feels different from the rest of the breast tissue. More than 80% of cases are discovered when a person detects such a lump with the fingertips and there are various methods of assessing if the detected lump is a cyst, and if so, either benign or malignant.

One such method of detecting the dangerousness of the mass is Fine-needle Aspiration (FNA). This diagnostic procedure consists of a very safe and minor procedure, where a thin hollow needle is inserted into the mass for obtaining cell samples and analyzing them under a microscope. A major surgical biopsy can be avoided by performing FNA, which is safer and far less traumatic, and possibly eliminating the need for hospitalization and other complications.

The problem presented in this report deals with finding out which features of a FNA are more relevant in diagnosing a patient's mammary lump as benign or malignant. The data used is the Breast Cancer Wisconsin (Diagnostic) Data Set, a data set whose features are computed from digitized images of FNAs of breast mass.

Exploratory Data Analysis

The breast cancer dataset consists of 569 FNA procedures each with 32 features. Ten real-valued features are computed for each cell nucleus:

a) radius
b) texture
c) perimeter
d) area
e) smoothness
f) compactness
g) concavity
h) concave points
i) symmetry
j) fractal dimension

From these, the mean, standard error and worst, are computed, thus making 10 features into 30. The other 2 features are the FNA ID number and the diagnosis, the target binary variable, which has values 'M' (malignant) or 'B' (benign).

```r
cancer_data = read.csv('cancer.csv')
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 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
## 6        0.2087                0.07613    0.3345     0.8902        2.217
##   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
## 6   27.19      0.007510        0.03345      0.03672           0.01137
##   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
## 6     0.02165             0.005082        15.47         23.75          103.40
##   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
## 6      741.6           0.1791            0.5249          0.5355
##   concave.points_worst symmetry_worst fractal_dimension_worst  X
## 1               0.2654         0.4601                 0.11890 NA
## 2               0.1860         0.2750                 0.08902 NA
## 3               0.2430         0.3613                 0.08758 NA
## 4               0.2575         0.6638                 0.17300 NA
## 5               0.1625         0.2364                 0.07678 NA
## 6               0.1741         0.3985                 0.12440 NA
```

```r
nrows <- nrow(cancer_data)
ncols <- ncol(cancer_data)
cat("Breast cancer dataset size:", nrows, "rows x", ncols, "cols")
```

```
## Breast cancer dataset size: 569 rows x 33 cols
```

As an addendum when talking about the amount of features the dataset has, it can be noticed that instead of the 32 features mentioned, there are really 33 columns, this is because the last column, called ‘X’, is a pointless column present filled with ‘N/A’ values. This column will be dropped.

```r
cancer_data$X <- NULL
cat("Dataset contains NULL values:", any(is.na(cancer_data)), ’\n’)
```

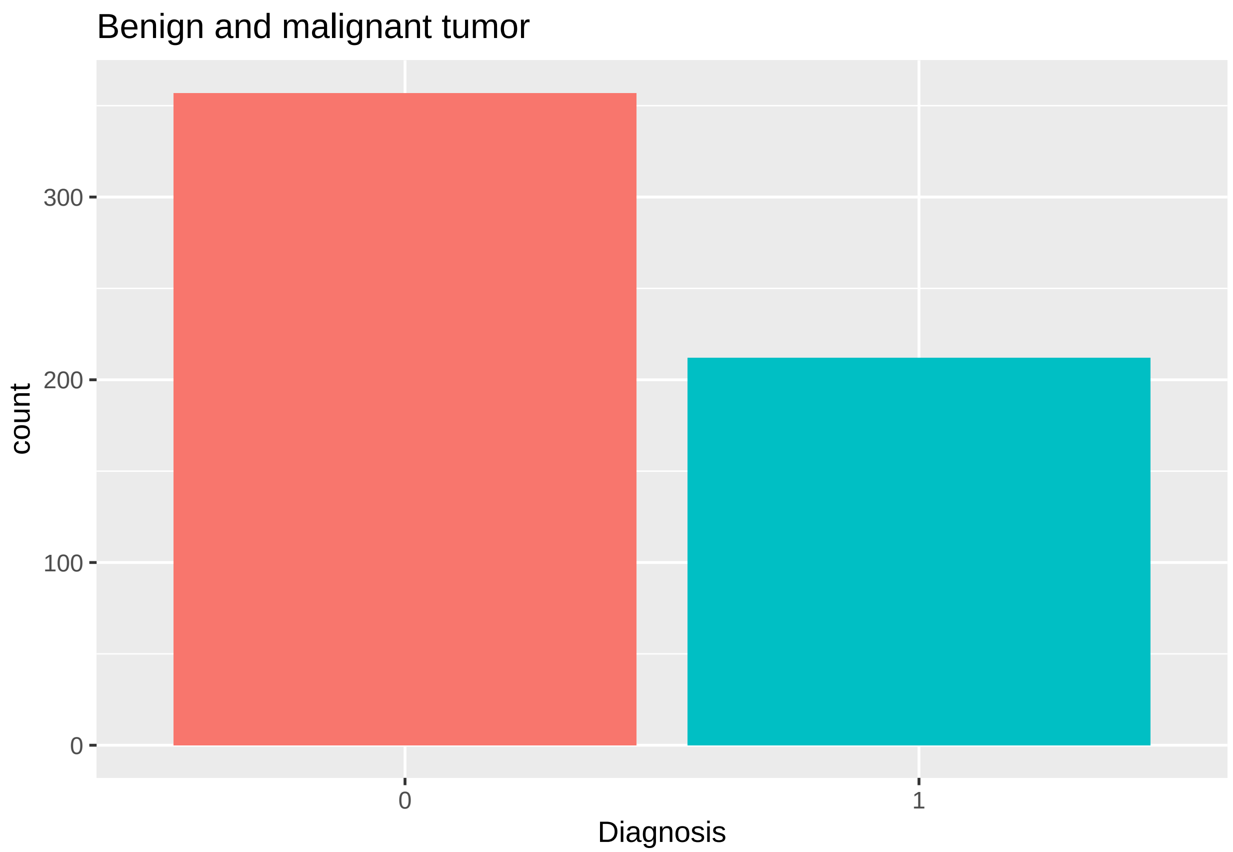
```
## Dataset contains NULL values: FALSE
```

In order to present our data into an numeric format, we convert the target output into binary values (B into ‘0’ and M into ‘1’)

```r
cancer_data[cancer_data == "B"] <- as.numeric(0)
cancer_data[cancer_data == "M"] <- as.numeric(1)
cancer_data$diagnosis <- as.numeric(as.character(cancer_data$diagnosis))
```

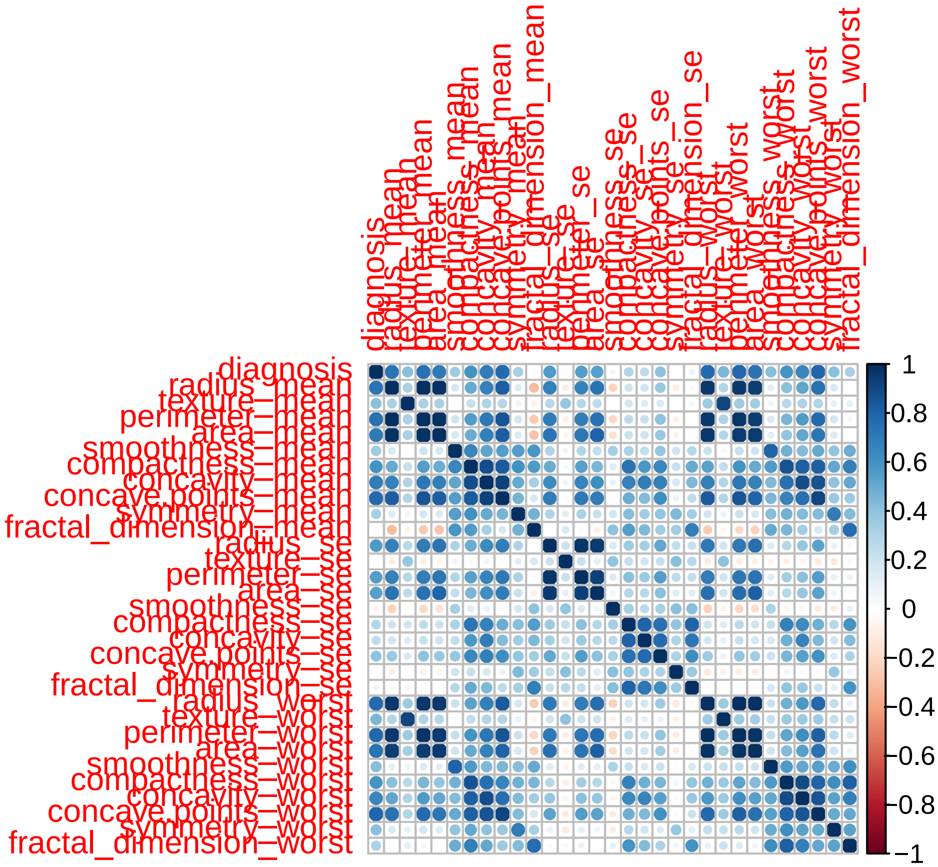
Is our data balanced? We would like to know more about how many benign and malignant tumors are contained in our dataset by visualizing a barplot.

```r
ggplot(cancer_data, aes(factor(diagnosis), fill=as.factor(diagnosis))) +
      geom_bar() + theme(legend.position="none") + xlab("Diagnosis") +
      ggtitle("Benign and malignant tumor")
```



A correlation matrix is visualized to obtain correlation coefficients between variables.

```
corrplot(cor(cancer_data[,2:32]))
```



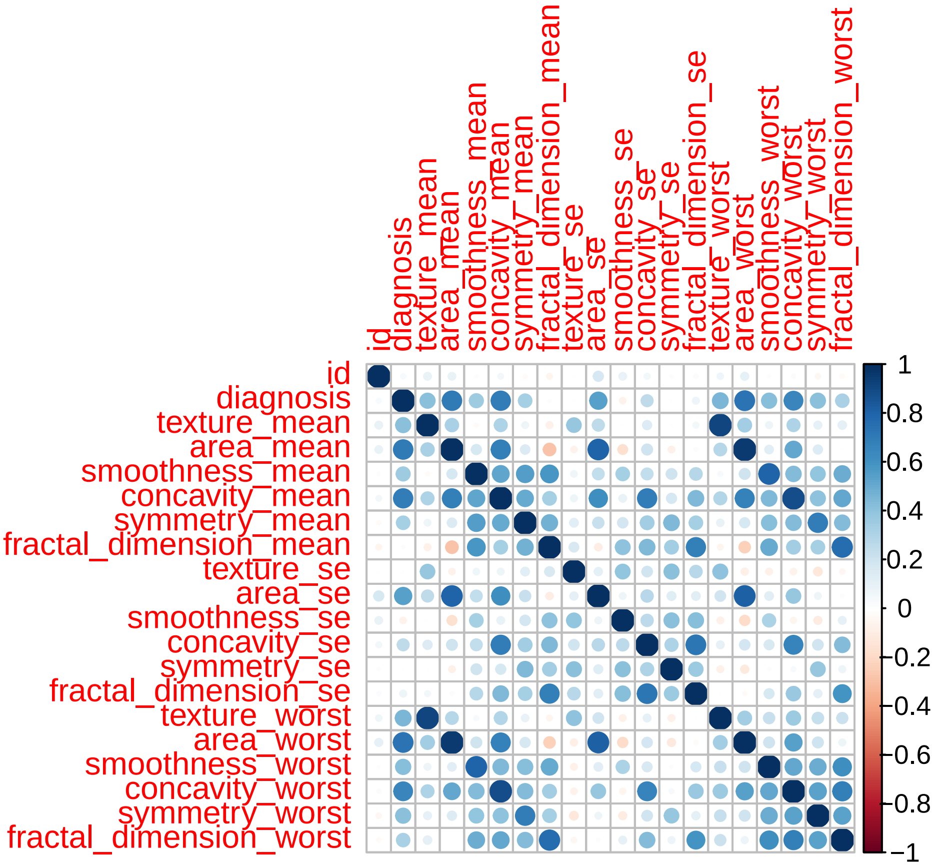
As seen in the correlation matrix, there are some variables that are *highly* correlated. Since they do not provide any new information, we could apply feature selection. For instance, `radius_mean`, `perimeter_mean` and `area_mean` are highly correlated, so we decide to keep `area_mean` in our dataset. Moreover, `compactness_mean`, `concavity_mean` and `concave.points_mean` are correlated, so we decide to keep `concavity_mean`. This process is also carried out for the standard deviation and worst related variables.

```r
# Dropping columns using the subset function. '-' indicates dropping vars.
cancer_data = subset(cancer_data, select = -c(radius_mean, perimeter_mean,
                       compactness_mean, concave.points_mean, radius_se,
                       perimeter_se, compactness_se, concave.points_se,
                       radius_worst, perimeter_worst, compactness_worst,
                       concave.points_worst))

variables <- c("texture_mean", "area_mean", "smoothness_mean", "concavity_mean",
                "symmetry_mean", "fractal_dimension_mean", "texture_se",
                "area_se", "smoothness_se", "concavity_se", "symmetry_se",
                "fractal_dimension_se", "texture_worst", "area_worst",
                "smoothness_worst", "concavity_worst", "symmetry_worst",
                "fractal_dimension_worst")
```

By performing this feature selection, we drastically removed the number of variables, from 30 to 18. The final variables, along with the correlation, used for our models are as follows:

```
corrplot(cor(cancer_data))
```



There are still some variables that are *slightly* correlated, but a more curated feature selection will be made later on in order to create our models.

Now, we would like to **scale** the columns of our data into the range [0,1] for easier comparison, except the diagnosis and id column. Thus, each column has a mean of 0 with a standard deviation of 1.

```
scaled_cancer_data <- cancer_data
col_names <- colnames(cancer_data) # Keep column names

scaled_cancer_data[,1:2] <- cancer_data[,1:2] # Id and diagnosis without scaling
scaled_cancer_data[,3:18] <- scale(cancer_data[,3:18])
colnames(scaled_cancer_data) <- col_names # Retrieve column names
head(scaled_cancer_data)
```

```
##         id diagnosis texture_mean  area_mean smoothness_mean concavity_mean
## 1   842302         1   -2.0715123  0.9835095       1.5670875     2.65054179
## 2   842517         1   -0.3533215  1.9070303      -0.8262354    -0.02382489
## 3 84300903         1    0.4557859  1.5575132       0.9413821     1.36227979
## 4 84348301         1    0.2535091 -0.7637917       3.2806668     1.91421287
## 5 84358402         1   -1.1508038  1.8246238       0.2801253     1.36980615
## 6   843786         1   -0.8346009 -0.5052059       2.2354545     0.86554001
##   symmetry_mean fractal_dimension_mean texture_se    area_se smoothness_se
## 1   2.215565542              2.2537638 -0.5647681  2.4853907    -0.2138135
## 2   0.001391139             -0.8678888 -0.8754733  0.7417493    -0.6048187
## 3   0.938858720             -0.3976580 -0.7793976  1.1802975    -0.2967439
## 4   2.864862154              4.9066020 -0.1103120 -0.2881246     0.6890953
## 5  -0.009552062             -0.5619555 -0.7895490  1.1893103     1.4817634
## 6   1.004517928              1.8883435 -0.5921406 -0.2890039     0.1562093
##   concavity_se symmetry_se fractal_dimension_se texture_worst area_worst
## 1    0.7233897   1.1477468           0.90628565   -1.35809849  1.9994782
## 2   -0.4403926  -0.8047423          -0.09935632   -0.36887865  1.8888270
## 3    0.2128891   0.2368272           0.29330133   -0.02395331  1.4550043
## 4    0.8187979   4.7285198           2.04571087    0.13386631 -0.5495377
## 5    0.8277425  -0.3607748           0.49888916   -1.46548091  1.2196511
## 6    0.1598845   0.1340009           0.48641784   -0.31356043 -0.2441054
##   smoothness_worst concavity_worst symmetry_worst fractal_dimension_worst
## 1        1.3065367       2.1076718         0.4601                 0.11890
## 2       -0.3752817      -0.1466200         0.2750                 0.08902
## 3        0.5269438       0.8542223         0.3613                 0.08758
## 4        3.3912907       1.9878392         0.6638                 0.17300
## 5        0.2203623       0.6126397         0.2364                 0.07678
## 6        2.0467119       1.2621327         0.3985                 0.12440
```

Main Modeling Idea

As a summary, our modeling idea has been to do the following analysis' with different types of models:

1. **Linear Model**, with each of the remaining 18 features, to see which individual variable might have more influence in predicting correct diagnosis.
2. **Multivariate Models**, done with 3 blocks of 6 variables from mean, se and worst, to see which variable block might have more influence in predicting correct diagnosis.
3. **Multivariate Model**, done with all 18 features in order to obtain proper posterior checking, and see the minimum optimal amount of variables needed, and which ones, for equal or better prediction as this model with all 18 features.
4. **Multivariate Model**, done with the best minimum optimal amount of variables, to check if the predictions obtained are equal or better than the model with all variables.
5. **Multivariate Model**, done with Regularized Horseshoe Prior with the best minimum optimal amount of variables, in order to check if the predictions obtained are equal or better than the model with all variables.
6. **Gaussian Model**, . . .

Methodology

Priors

We have chosen to work with Weakly Informative Priors because:

1. A weakly informative prior means a reasonable representation of partial ignorance about the data, but that still includes a small amount of real-world information. Uniformity of distribution on an appropriate measurement scale means that the prior does not strongly favor particular values of the parameter.

2. A weakly informative prior does not contribute strongly to the posterior. With a weakly informative prior we say that this small contribution from the prior "lets the data speak for itself".

Because the target feature, the diagnosis, has binary values we will use a Bernoulli logistic regression approach:

$$y \sim Bernoulli(y \mid logit^{-1}(\alpha + \beta \times x)),$$

with α and β are the intercept and regression coefficients, x are the predictor variables and y is the breast cancer diagnosis.

Coefficients

For the α intercept coefficient and β regression coefficient we have chosen a Normal and a Student T distribution respectively, to be more precise:

1. $\alpha \sim Normal(\mu_\alpha,\ \sigma_\alpha)$, with $\mu_\alpha = 0$ and $\sigma_\alpha = 1$.
2. $\beta \sim StudentT(df_\beta,\ location_\beta,\ scale_\beta)$, with $df_\beta = 3$, $location_\beta = 0$ and $scale_\beta = 1$.

The reason behind this choice in priors is because we scale our data, so it makes sense to select a normal distribution with mean centered in 0 and standard deviation 1.

```r
beta <- student_t(df = 3, location = 0, scale = 1)
alpha <- normal(0, 1)
```

Regularized Horseshoe Prior

We have also used a Weakly Regularized Horseshoe Prior, as *Piironen+Vehtari:RHS:2017*, where we included the prior assumption that a number of variables are more relevant that the rest. The assumption origin will be discussed in a future section, but as a summary, it has been assumed that there are 3 relevant variables, while the others are irrelevant.

$$PONERLASFORMULASAQUI$$

```r
p0 <- 3 # number of relevant variables
tau0 <- p0 / (length(variables) - p0) * 1 / sqrt(nrows)
rhs_prior <- hs(global_scale=tau0)
```

Linear Models

Multivariate Models - Variable Blocks

Multivariate Model - Full Model

Multivariate Model -

Results

Conclusion

References