Final Project Report

Olivia Fan, Alicia Gong

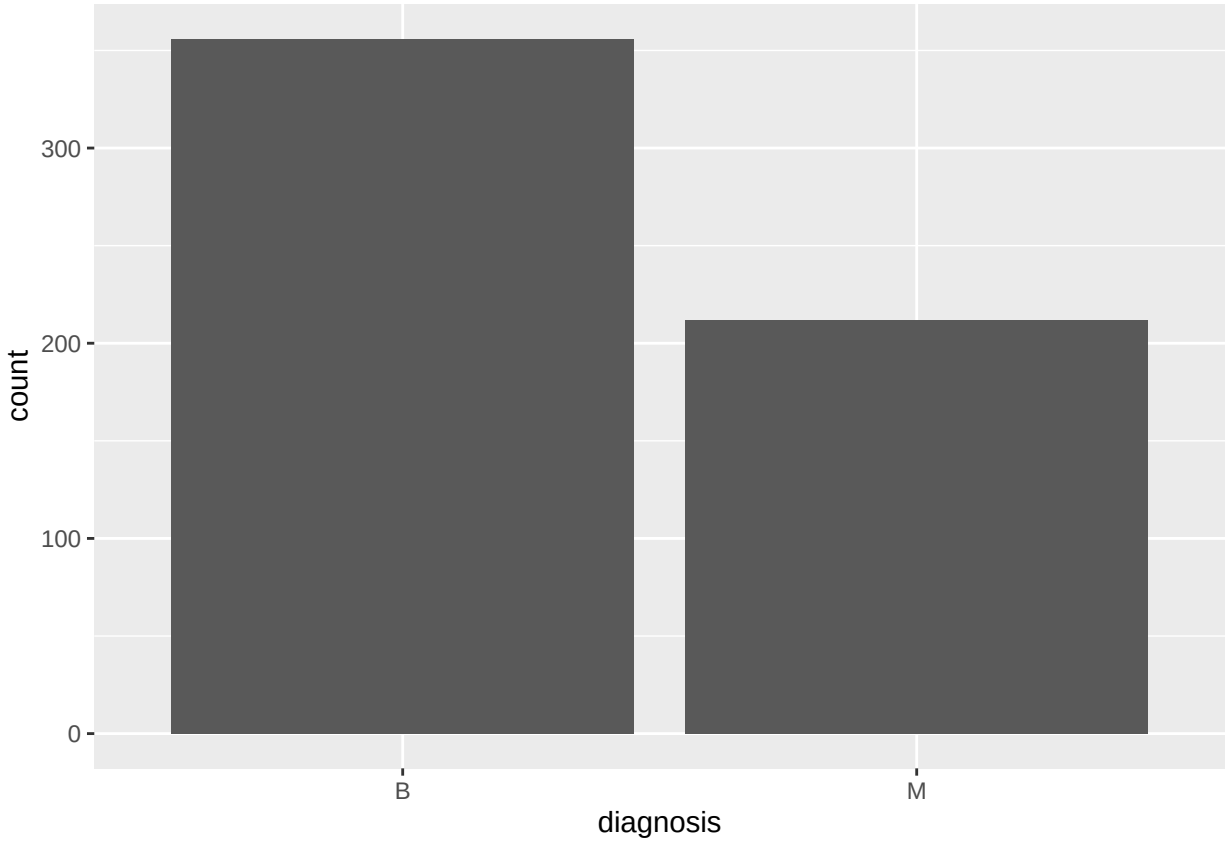
06 December, 2022

Data Processing

In order to fit SVM on the data, we encode the `diagnosis` variable into a factor variable with level 1 and -1:

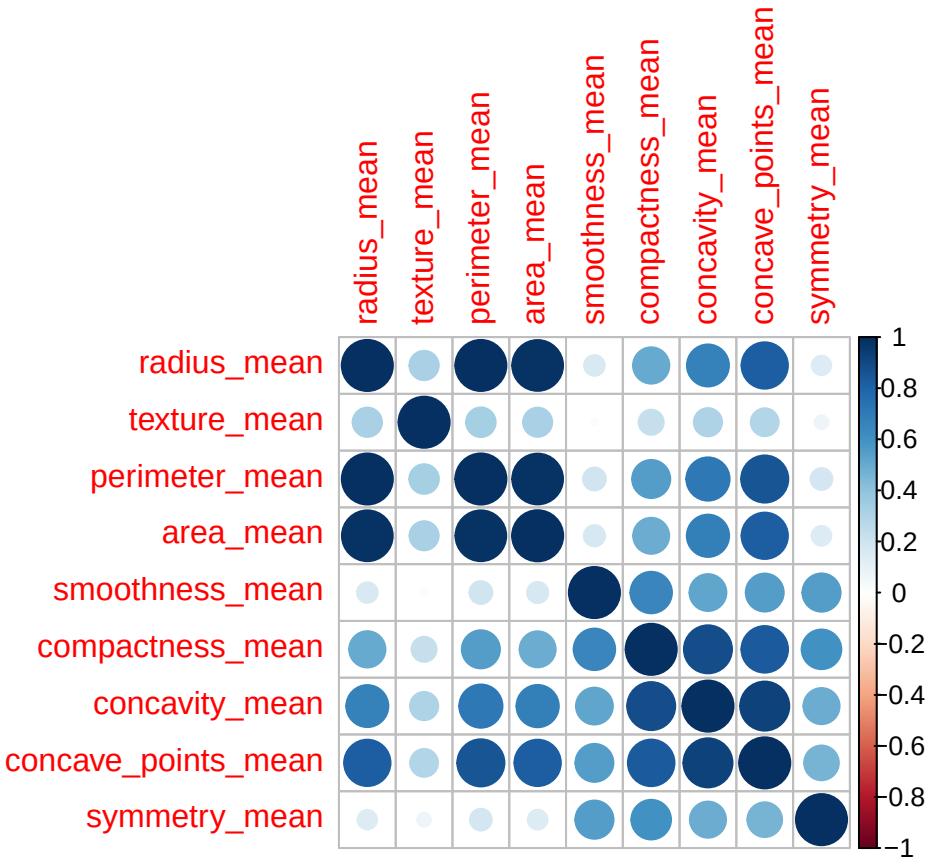
We partition the data into training and testing sets using a 70-30 percentage split(70% of the original data as the training set, and 30% as the testing set):

EDA



The bar plot shows that there is a larger number of benign than malignant cancer.

We divide the data into 3 categories according to their features.



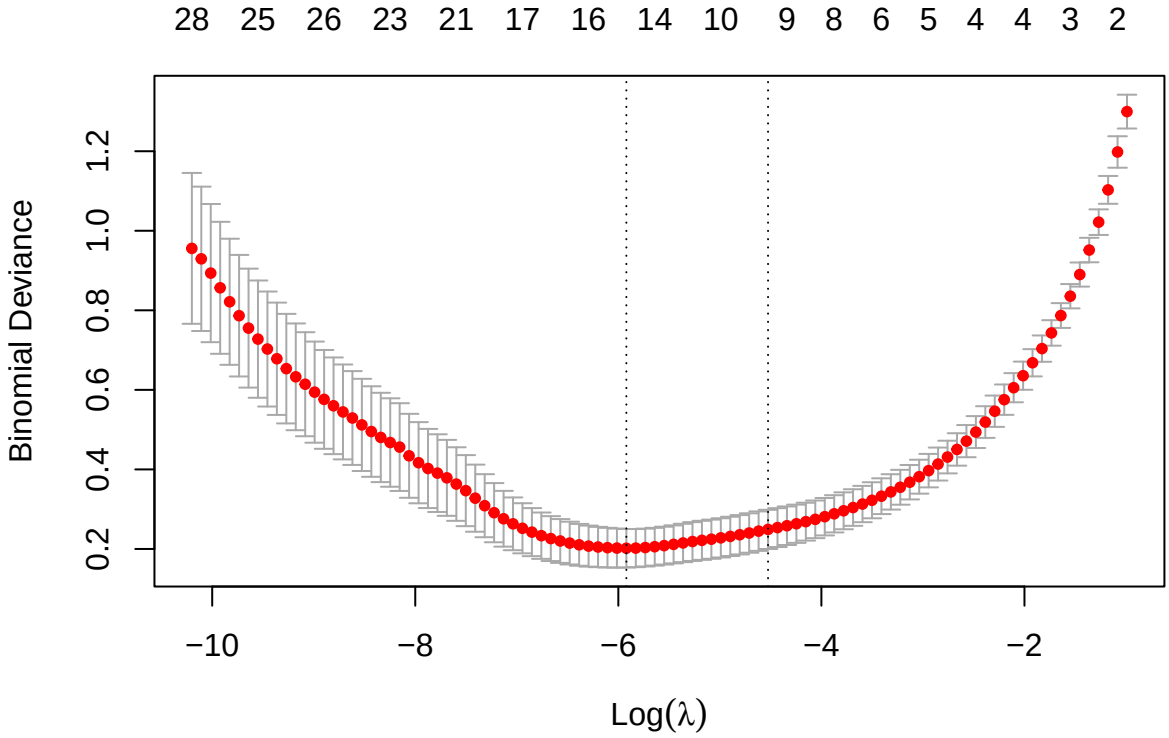
Major observations:

- Radius_mean, perimeter_mean, and area_mean are highly correlated.
- Compactness_mean, concavity_mean and concave_points_mean are highly correlated.

Methodology

SVM

Model Selection (Lasso penalized logistic regression)




```
## [1] 0.002682998
```

```
## 31 x 1 sparse Matrix of class "dgCMatrix"
##                                  s0
## (Intercept)             -28.77675317
## radius_mean               .
## texture_mean              0.05593184
## perimeter_mean            .
## area_mean                 .
## smoothness_mean           .
## compactness_mean          .
## concavity_mean            .
## concave_points_mean      26.70558695
## symmetry_mean             .
## fractal_dimension_mean    .
## radius_se                 4.68352247
## texture_se               -0.53071651
## perimeter_se              .
## area_se                   0.04732961
## smoothness_se            88.30404283
## compactness_se          -42.98049312
## concavity_se              .
## concave_points_se         .
## symmetry_se               .
## fractal_dimension_se    -85.32165947
## radius_worst              0.58890672
## texture_worst             0.22994413
## perimeter_worst           .
## area_worst                .
```

```
## smoothness_worst         17.82352005
## compactness_worst         .
## concavity_worst           4.35034427
## concave_points_worst     21.35593118
## symmetry_worst            7.99935194
## fractal_dimension_worst   .
```

Linear Kernel SVM

We use the predictors selected by the LASSO penalized logistic regression as predictors for the support vector machine model:

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost
##   0.1
##
## - best performance: 0.03282051
##
## - Detailed performance results:
##    cost      error dispersion
## 1 1e-03 0.10576923 0.04378205
## 2 1e-02 0.04794872 0.03015895
## 3 1e-01 0.03282051 0.02914580
## 4 1e+00 0.04282051 0.02652294
## 5 5e+00 0.04282051 0.03132482
## 6 1e+01 0.03769231 0.03164962
## 7 1e+02 0.03775641 0.02700681
```

```
##
## Call:
## best.tune(METHOD = svm, train.x = diagnosis_binary ~ concavity_mean +
##     concave_points_mean + radius_se + texture_se + smoothness_se +
##     compactness_se + fractal_dimension_se + radius_worst + texture_worst +
##     smoothness_worst + concavity_worst + concave_points_worst + symmetry_worst +
##     fractal_dimension_worst, data = cancer_train, ranges = list(cost = c(0.001,
##     0.01, 0.1, 1, 5, 10, 100)), kernel = "linear")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  0.1
##
## Number of Support Vectors:  62
##
##  ( 30 32 )
##
##
## Number of Classes:  2
##
```

```
## Levels:
##  -1 1
```

```
##        truth
## predict -1  1
##      -1 97  2
##      1   1 71
```

```
## [1] 0.01754386
```

The misclassification rate is 0.0467.

Radial Kernel SVM

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  cost gamma
##     1   0.5
##
## - best performance: 0.04794872
##
## - Detailed performance results:
##     cost gamma      error dispersion
## 1  1e-01   0.5 0.35019231 0.06877558
## 2  1e+00   0.5 0.04794872 0.03641743
## 3  1e+01   0.5 0.05544872 0.02849221
## 4  1e+02   0.5 0.05544872 0.02849221
## 5  1e+03   0.5 0.05544872 0.02849221
## 6  1e-01   1.0 0.35019231 0.06877558
## 7  1e+00   1.0 0.20423077 0.07710401
## 8  1e+01   1.0 0.17397436 0.07285805
## 9  1e+02   1.0 0.17397436 0.07285805
## 10 1e+03   1.0 0.17397436 0.07285805
## 11 1e-01   2.0 0.35019231 0.06877558
## 12 1e+00   2.0 0.34262821 0.05486857
## 13 1e+01   2.0 0.33506410 0.05581714
## 14 1e+02   2.0 0.33506410 0.05581714
## 15 1e+03   2.0 0.33506410 0.05581714
## 16 1e-01   3.0 0.35019231 0.06877558
## 17 1e+00   3.0 0.35019231 0.06877558
## 18 1e+01   3.0 0.35019231 0.06877558
## 19 1e+02   3.0 0.35019231 0.06877558
## 20 1e+03   3.0 0.35019231 0.06877558
## 21 1e-01   4.0 0.35019231 0.06877558
## 22 1e+00   4.0 0.35019231 0.06877558
## 23 1e+01   4.0 0.35019231 0.06877558
## 24 1e+02   4.0 0.35019231 0.06877558
## 25 1e+03   4.0 0.35019231 0.06877558
```

```
##
## Call:
## best.tune(METHOD = svm, train.x = diagnosis_binary ~ concavity_mean +
```

```
##     concave_points_mean + radius_se + texture_se + smoothness_se +
##     compactness_se + fractal_dimension_se + radius_worst + texture_worst +
##     smoothness_worst + concavity_worst + concave_points_worst + symmetry_worst +
##     fractal_dimension_worst, data = cancer_train, ranges = list(cost = c(0.1,
##     1, 10, 100, 1000), gamma = c(0.5, 1, 2, 3, 4)), kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  278
##
##  ( 128 150 )
##
##
## Number of Classes:  2
##
## Levels:
##  -1 1
```

```
##        truth
## predict -1  1
##      -1 94  3
##      1   4 70
```

Random Forest

```
##
## Call:
##  randomForest(formula = diagnosis_binary ~ texture_mean + perimeter_mean +      smoothness_mean + co
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 8.56%
## Confusion matrix:
##     -1   1 class.error
## -1 246  12  0.04651163
## 1   22 117  0.15827338
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction -1  1
##         -1 97 10
##         1   1 63
##
##                Accuracy : 0.9357
##                  95% CI : (0.8878, 0.9675)
##     No Information Rate : 0.5731
##     P-Value [Acc > NIR] : < 2e-16
##
```

```
##                  Kappa : 0.8664
##
##  Mcnemar's Test P-Value : 0.01586
##
##             Sensitivity : 0.9898
##             Specificity : 0.8630
##          Pos Pred Value : 0.9065
##          Neg Pred Value : 0.9844
##              Prevalence : 0.5731
##          Detection Rate : 0.5673
##    Detection Prevalence : 0.6257
##       Balanced Accuracy : 0.9264
##
##        'Positive' Class : -1
##
```