1. INTRODUCTION

1. INTRODUCTION

In this project, we use breast cancer biopsy data provided by UCI to build a classification machine learning model with the aim of distinguishing between a malignant and benign breast mass.

The dataset consists of two outcomes in the y label:

- "M" denoting a malignant breast mass (cancer detected)
- "B" denoting a benign breast mass (no cancer detected)

And predictors in the x label consisting of means, standard errors and worst values of each 10 nuclear measurements. A total of 30 features per biopsy.

The steps followed include:

- Preparation of the work environment.
- Preparation, exploration and visualizations of the data.
- Analysis of the predictors
- Model selection.

2. METHODS AND ANALYSIS

2.1 Work Environment and Data Preparation

We are going to use the following libraries:

```
if(!require(tidyverse)) install.packages("tidyverse",
                                         repos = "http://cran.us.r-project.org")
if(!require(dslabs)) install.packages("dslabs",
                                      repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret",
                                      repos = "http://cran.us.r-project.org")
if(!require(ggplot2)) install.packages("ggplot2",
```

```
                                        repos = "http://cran.us.r-project.org")
if(!require(knitr)) install.packages("knitr",
                                     repos = "http://cran.us.r-project.org")
if(!require(rmarkdown)) install.packages("rmarkdown",
                                         repos = "http://cran.us.r-project.org")
```

The dataset we are going to use to train our model is contained in the dslabs package so we shall proceed by loading the data using the code

```
data(brca)
```

2.2 Data Exploration, Analysis and Visualizations

The dataset contains predictors(x) and outcomes(y), where:

The outcomes are

```
unique(brca$y)
```

```
## [1] B M
## Levels: B M
```

with **357** benign observations and **212** malignant observations

```
table(brca$y)
```

```
##
##   B   M
## 357 212
```

And the predictors which are in a matrix consist of the mean, standard error and worst value of 10 nuclear measurements on the slide per biopsy

```
head(brca$x)
```

```
##      radius_mean texture_mean perimeter_mean area_mean smoothness_mean
## [1,]      13.540        14.36          87.46     566.3         0.09779
## [2,]      13.080        15.71          85.63     520.0         0.10750
## [3,]       9.504        12.44          60.34     273.9         0.10240
## [4,]      13.030        18.42          82.61     523.8         0.08983
## [5,]       8.196        16.84          51.71     201.9         0.08600
## [6,]      12.050        14.63          78.04     449.3         0.10310
##      compactness_mean concavity_mean concave_pts_mean symmetry_mean
```

```
## [1,]          0.08129        0.06664         0.047810        0.1885
## [2,]          0.12700        0.04568         0.031100        0.1967
## [3,]          0.06492        0.02956         0.020760        0.1815
## [4,]          0.03766        0.02562         0.029230        0.1467
## [5,]          0.05943        0.01588         0.005917        0.1769
## [6,]          0.09092        0.06592         0.027490        0.1675
##      fractal_dim_mean radius_se texture_se perimeter_se area_se
## [1,]          0.05766    0.2699     0.7886        2.058  23.560
## [2,]          0.06811    0.1852     0.7477        1.383  14.670
## [3,]          0.06905    0.2773     0.9768        1.909  15.700
## [4,]          0.05863    0.1839     2.3420        1.170  14.160
## [5,]          0.06503    0.1563     0.9567        1.094   8.205
## [6,]          0.06043    0.2636     0.7294        1.848  19.870
##      smoothness_se compactness_se concavity_se concave_pts_se symmetry_se
## [1,]      0.008462       0.014600      0.02387       0.013150     0.01980
## [2,]      0.004097       0.018980      0.01698       0.006490     0.01678
## [3,]      0.009606       0.014320      0.01985       0.014210     0.02027
## [4,]      0.004352       0.004899      0.01343       0.011640     0.02671
## [5,]      0.008968       0.016460      0.01588       0.005917     0.02574
## [6,]      0.005488       0.014270      0.02322       0.005660     0.01428
##      fractal_dim_se radius_worst texture_worst perimeter_worst area_worst
## [1,]       0.002300       15.110         19.26           99.70      711.2
## [2,]       0.002425       14.500         20.49           96.09      630.5
## [3,]       0.002968       10.230         15.66           65.13      314.9
## [4,]       0.001777       13.300         22.81           84.46      545.9
## [5,]       0.002582        8.964         21.96           57.26      242.2
## [6,]       0.002422       13.760         20.70           89.88      582.6
##      smoothness_worst compactness_worst concavity_worst concave_pts_worst
## [1,]          0.14400           0.17730         0.23900           0.12880
## [2,]          0.13120           0.27760         0.18900           0.07283
## [3,]          0.13240           0.11480         0.08867           0.06227
## [4,]          0.09701           0.04619         0.04833           0.05013
## [5,]          0.12970           0.13570         0.06880           0.02564
## [6,]          0.14940           0.21560         0.30500           0.06548
##      symmetry_worst fractal_dim_worst
## [1,]         0.2977           0.07259
## [2,]         0.3184           0.08183
## [3,]         0.2450           0.07773
## [4,]         0.1987           0.06169
## [5,]         0.3105           0.07409
## [6,]         0.2747           0.08301
```

A total of **569** observations and **30** features

```
dim(brca$x)
```

```
## [1] 569  30
```

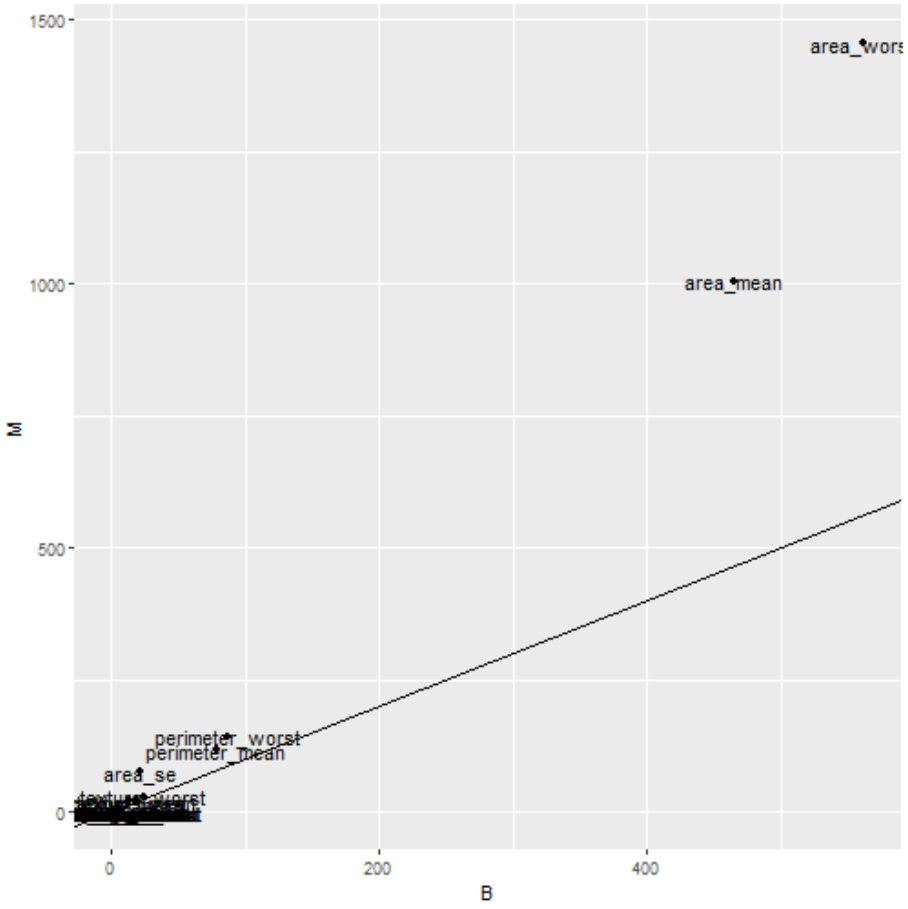
Before proceeding, we split our data into training and test sets. In this project, we shall use 80% of the data to train our model and the remaining to test our final model

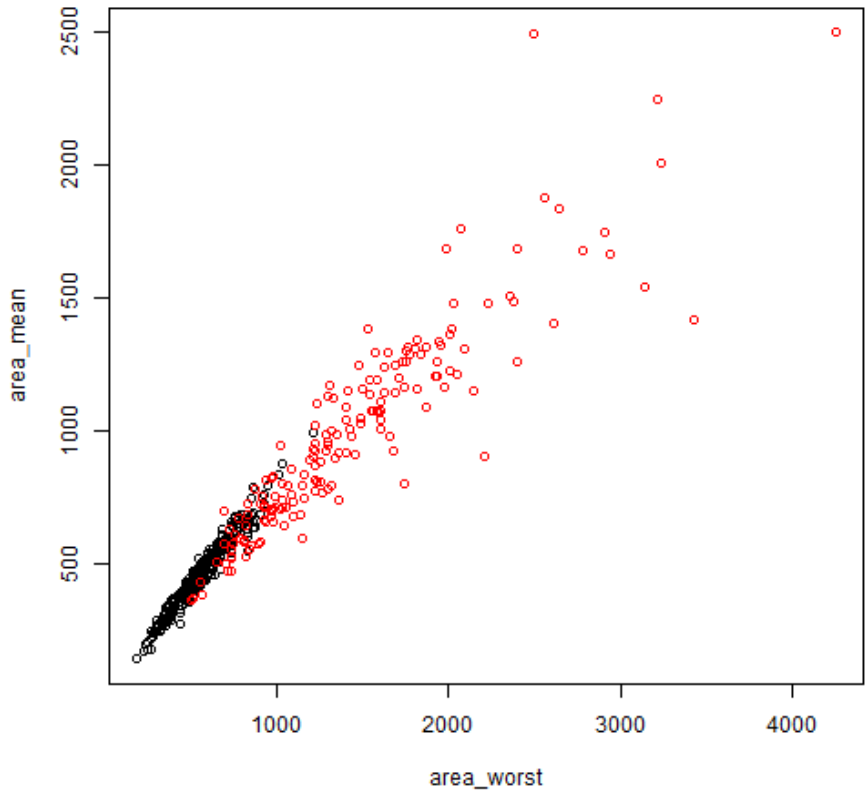
```
set.seed(1)
test_index <- createDataPartition(y =  brca$y, times = 1, p = 0.2, list = FALSE)
```

```
y <- droplevels(brca$y[-test_index])
x <- brca$x[-test_index, ]
brca_train_set <- data.frame(x,y)
write.csv(brca_train_set, file = "data/brca_train_set.csv")

y2 <- droplevels(brca$y[test_index])
x2 <- brca$x[test_index, ]
brca_test_set <- data.frame(x2,y2)
write.csv(brca_test_set, file = "data/brca_test_set.csv")
```

From the visualization below, we can conclude that **area_worst** and **area_mean** are the two features driving our algorithm.


1500
1000
500
0
M
0
200
400
B
area_wors
area_mean
perimeter_worst
perimeter_mean
area_se



2.3 Data Modelling

From the data visualizations and observations, it is evident that this is a categorical outcome since y can be malignant or benign with 30 predictors. We will therefore fit a linear model applying a model ensemble and select the best performing based on accuracy. For this, we shall use the caret package which is already preloaded.

```
# Apply model ensemble

models <- c("glm", "lda", "naive_bayes", "svmLinear", "knn",
            "gamLoess", "multinom", "qda", "rf", "adaboost")

fits <- lapply(models, function(model){
  train(y ~ ., method = model, data = brca_train_set)
```

```
  })
```

```
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 88.684873
## iter  20 value 21.522717
## iter  30 value 18.379848
## iter  40 value 16.880655
## iter  50 value 15.560688
## iter  60 value 13.273944
## iter  70 value 12.552922
## iter  80 value 11.806085
## iter  90 value 10.922810
## iter 100 value 10.582913
## final  value 10.582913
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 92.681433
## iter  20 value 38.930579
## iter  30 value 36.102419
## iter  40 value 35.899922
## final  value 35.899006
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 88.689347
## iter  20 value 21.749688
## iter  30 value 18.759545
## iter  40 value 17.389731
## iter  50 value 16.203981
## iter  60 value 14.902083
## iter  70 value 14.607084
## iter  80 value 14.241884
## iter  90 value 14.067388
## iter 100 value 13.849950
## final  value 13.849950
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 85.513723
## iter  20 value 8.871364
## iter  30 value 0.440395
## iter  40 value 0.001314
## final  value 0.000080
## converged
```

```
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 89.143851
## iter  20 value 30.422667
## iter  30 value 27.576088
## iter  40 value 27.417130
## final  value 27.416789
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 85.517580
## iter  20 value 9.481829
## iter  30 value 4.731543
## iter  40 value 3.395700
## iter  50 value 3.167434
## iter  60 value 3.006933
## iter  70 value 2.646251
## iter  80 value 2.475773
## iter  90 value 2.336107
## iter 100 value 2.193922
## final  value 2.193922
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 81.711480
## iter  20 value 21.837141
## iter  30 value 16.021581
## iter  40 value 12.627116
## iter  50 value 10.275096
## iter  60 value 6.751182
## iter  70 value 0.627756
## iter  80 value 0.255530
## iter  90 value 0.175752
## iter 100 value 0.160209
## final  value 0.160209
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 84.100275
## iter  20 value 37.731741
## iter  30 value 32.530685
## iter  40 value 32.057549
## iter  50 value 32.040633
## final  value 32.040632
## converged
## # weights:  32 (31 variable)
```

```
## initial  value 314.688820
## iter  10 value 81.714048
## iter  20 value 22.097956
## iter  30 value 16.721907
## iter  40 value 14.086778
## iter  50 value 12.412272
## iter  60 value 11.169713
## iter  70 value 10.636796
## iter  80 value 10.130877
## iter  90 value 9.410879
## iter 100 value 9.198045
## final  value 9.198045
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 78.334877
## iter  20 value 26.403745
## iter  30 value 21.888149
## iter  40 value 18.002807
## iter  50 value 16.346204
## iter  60 value 14.494966
## iter  70 value 13.869203
## iter  80 value 13.208164
## iter  90 value 12.940529
## iter 100 value 12.184202
## final  value 12.184202
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 83.279615
## iter  20 value 42.159774
## iter  30 value 37.441779
## iter  40 value 37.430334
## final  value 37.430323
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 78.340161
## iter  20 value 26.548204
## iter  30 value 22.286487
## iter  40 value 18.991956
## iter  50 value 17.828280
## iter  60 value 17.066395
## iter  70 value 16.711326
## iter  80 value 16.577566
## iter  90 value 16.231570
```

```
## iter 100 value 16.170547
## final  value 16.170547
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 85.842714
## iter  20 value 35.263236
## iter  30 value 23.710867
## iter  40 value 18.739239
## iter  50 value 15.532638
## iter  60 value 12.879941
## iter  70 value 10.348374
## iter  80 value 9.225765
## iter  90 value 6.709977
## iter 100 value 5.701080
## final  value 5.701080
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 95.099602
## iter  20 value 49.866787
## iter  30 value 47.759857
## iter  40 value 47.688546
## final  value 47.688526
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 85.852529
## iter  20 value 35.475380
## iter  30 value 25.152110
## iter  40 value 21.330441
## iter  50 value 19.369329
## iter  60 value 18.053445
## iter  70 value 17.529081
## iter  80 value 17.293254
## iter  90 value 17.150490
## iter 100 value 16.785195
## final  value 16.785195
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 73.115398
## iter  20 value 25.038836
## iter  30 value 17.234329
## iter  40 value 9.919557
## iter  50 value 7.545606
```

```
## iter  60 value 2.203448
## iter  70 value 0.108270
## iter  80 value 0.062981
## iter  90 value 0.023897
## iter 100 value 0.020657
## final  value 0.020657
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 79.699947
## iter  20 value 38.911732
## iter  30 value 32.512457
## iter  40 value 30.212755
## iter  50 value 30.164775
## final  value 30.164775
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 73.122775
## iter  20 value 25.242193
## iter  30 value 18.133413
## iter  40 value 13.924443
## iter  50 value 10.858772
## iter  60 value 10.448774
## iter  70 value 10.013114
## iter  80 value 9.767583
## iter  90 value 9.431063
## iter 100 value 9.003462
## final  value 9.003462
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 87.413432
## iter  20 value 36.390315
## iter  30 value 28.411661
## iter  40 value 17.509423
## iter  50 value 11.497598
## iter  60 value 6.523142
## iter  70 value 0.881206
## iter  80 value 0.547586
## iter  90 value 0.419896
## iter 100 value 0.344758
## final  value 0.344758
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
```

```
## iter  10 value 93.551564
## iter  20 value 57.170397
## iter  30 value 45.958872
## iter  40 value 42.873489
## iter  50 value 42.862276
## iter  50 value 42.862276
## iter  50 value 42.862276
## final  value 42.862276
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 87.420063
## iter  20 value 36.606298
## iter  30 value 28.794596
## iter  40 value 21.466706
## iter  50 value 19.071544
## iter  60 value 18.559232
## iter  70 value 18.309416
## iter  80 value 17.912021
## iter  90 value 17.217964
## iter 100 value 16.191542
## final  value 16.191542
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 73.982226
## iter  20 value 15.245689
## iter  30 value 9.500635
## iter  40 value 7.807714
## iter  50 value 5.798581
## iter  60 value 0.641990
## iter  70 value 0.005986
## iter  80 value 0.001452
## iter  90 value 0.001146
## iter 100 value 0.001002
## final  value 0.001002
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 78.607881
## iter  20 value 38.970014
## iter  30 value 30.740560
## iter  40 value 30.735278
## iter  50 value 30.734125
## final  value 30.734125
## converged
```

```
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 73.987207
## iter  20 value 15.722503
## iter  30 value 11.916893
## iter  40 value 10.451526
## iter  50 value 8.492540
## iter  60 value 7.978049
## iter  70 value 7.609952
## iter  80 value 7.006186
## iter  90 value 6.658560
## iter 100 value 6.497422
## final  value 6.497422
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 84.239155
## iter  20 value 12.630489
## iter  30 value 1.958976
## iter  40 value 0.005855
## final  value 0.000028
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 87.083733
## iter  20 value 29.619503
## iter  30 value 26.457802
## iter  40 value 26.427227
## iter  50 value 26.425939
## iter  50 value 26.425938
## iter  50 value 26.425938
## final  value 26.425938
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 84.242247
## iter  20 value 13.044675
## iter  30 value 8.013552
## iter  40 value 6.117229
## iter  50 value 5.736345
## iter  60 value 5.396934
## iter  70 value 5.092803
## iter  80 value 4.881202
## iter  90 value 4.602075
## iter 100 value 4.192966
## final  value 4.192966
```

```
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 85.298456
## iter  20 value 31.108504
## iter  30 value 24.705030
## iter  40 value 20.124611
## iter  50 value 16.639558
## iter  60 value 14.663042
## iter  70 value 13.659723
## iter  80 value 12.936788
## iter  90 value 12.138430
## iter 100 value 11.701746
## final  value 11.701746
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 89.789477
## iter  20 value 53.466527
## iter  30 value 47.272827
## iter  40 value 47.263703
## final  value 47.263264
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 85.303342
## iter  20 value 31.408228
## iter  30 value 25.319581
## iter  40 value 21.946883
## iter  50 value 19.415776
## iter  60 value 18.371484
## iter  70 value 18.145537
## iter  80 value 17.821769
## iter  90 value 17.437989
## iter 100 value 16.938126
## final  value 16.938126
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 98.086831
## iter  20 value 21.015366
## iter  30 value 10.129206
## iter  40 value 3.446131
## iter  50 value 0.303648
## iter  60 value 0.000639
## final  value 0.000000
```

```
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 102.480200
## iter  20 value 42.106627
## iter  30 value 37.075402
## iter  40 value 36.875133
## final  value 36.875059
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 98.091509
## iter  20 value 21.444002
## iter  30 value 14.395481
## iter  40 value 11.924783
## iter  50 value 11.321104
## iter  60 value 10.951687
## iter  70 value 10.759741
## iter  80 value 10.631324
## iter  90 value 10.435594
## iter 100 value 10.262851
## final  value 10.262851
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 77.650180
## iter  20 value 24.823184
## iter  30 value 20.598095
## iter  40 value 17.636877
## iter  50 value 15.853783
## iter  60 value 14.952065
## iter  70 value 14.602631
## iter  80 value 14.429214
## iter  90 value 13.806279
## iter 100 value 12.740388
## final  value 12.740388
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 83.079703
## iter  20 value 40.078960
## iter  30 value 37.508496
## iter  40 value 37.475454
## final  value 37.475440
## converged
## # weights:  32 (31 variable)
```

```
## initial  value 314.688820
## iter  10 value 77.655954
## iter  20 value 25.138077
## iter  30 value 21.302374
## iter  40 value 18.869298
## iter  50 value 17.560414
## iter  60 value 17.080036
## iter  70 value 16.911123
## iter  80 value 16.763908
## iter  90 value 16.117410
## iter 100 value 15.968791
## final  value 15.968791
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 96.632247
## iter  20 value 28.249429
## iter  30 value 21.359665
## iter  40 value 18.015046
## iter  50 value 15.150891
## iter  60 value 13.400690
## iter  70 value 12.394376
## iter  80 value 11.598377
## iter  90 value 10.950660
## iter 100 value 9.052303
## final  value 9.052303
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 100.162669
## iter  20 value 42.277664
## iter  30 value 37.383351
## iter  40 value 37.351297
## final  value 37.351268
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 96.636002
## iter  20 value 28.434323
## iter  30 value 21.925820
## iter  40 value 19.204840
## iter  50 value 17.410923
## iter  60 value 15.361722
## iter  70 value 14.672995
## iter  80 value 14.475639
## iter  90 value 14.231335
```

```
## iter 100 value 14.018474
## final  value 14.018474
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 82.875512
## iter  20 value 22.914518
## iter  30 value 19.951213
## iter  40 value 18.321021
## iter  50 value 17.301046
## iter  60 value 16.141676
## iter  70 value 15.262011
## iter  80 value 14.487218
## iter  90 value 13.842072
## iter 100 value 12.758787
## final  value 12.758787
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 86.504544
## iter  20 value 35.622338
## iter  30 value 30.486984
## iter  40 value 30.468469
## iter  50 value 30.467767
## final  value 30.467767
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 82.879376
## iter  20 value 23.176598
## iter  30 value 20.510489
## iter  40 value 19.070927
## iter  50 value 18.292004
## iter  60 value 17.711174
## iter  70 value 17.321482
## iter  80 value 17.090859
## iter  90 value 16.921080
## iter 100 value 16.291816
## final  value 16.291816
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 94.502249
## iter  20 value 26.748991
## iter  30 value 20.718150
## iter  40 value 11.818949
```

```
## iter  50 value 9.397223
## iter  60 value 5.119066
## iter  70 value 2.231136
## iter  80 value 1.404211
## iter  90 value 1.039625
## iter 100 value 0.781740
## final  value 0.781740
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 98.683360
## iter  20 value 42.648717
## iter  30 value 40.696635
## iter  40 value 40.642216
## final  value 40.642172
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 94.506791
## iter  20 value 27.072769
## iter  30 value 21.491834
## iter  40 value 15.964443
## iter  50 value 15.374656
## iter  60 value 14.818563
## iter  70 value 14.436328
## iter  80 value 14.137915
## iter  90 value 13.919072
## iter 100 value 13.193164
## final  value 13.193164
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 90.879911
## iter  20 value 27.118673
## iter  30 value 23.455610
## iter  40 value 20.580996
## iter  50 value 19.647031
## iter  60 value 18.347100
## iter  70 value 17.658389
## iter  80 value 17.187197
## iter  90 value 16.795096
## iter 100 value 15.875427
## final  value 15.875427
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
```

```
## iter  10 value 92.167790
## iter  20 value 44.478558
## iter  30 value 40.704922
## iter  40 value 40.667433
## final  value 40.667409
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 90.886472
## iter  20 value 27.338762
## iter  30 value 23.885359
## iter  40 value 21.564041
## iter  50 value 20.964559
## iter  60 value 19.757034
## iter  70 value 19.383419
## iter  80 value 19.260179
## iter  90 value 19.077279
## iter 100 value 18.899435
## final  value 18.899435
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 69.873380
## iter  20 value 26.954752
## iter  30 value 22.283398
## iter  40 value 14.023157
## iter  50 value 12.163712
## iter  60 value 9.862000
## iter  70 value 8.161252
## iter  80 value 6.817130
## iter  90 value 5.019380
## iter 100 value 4.145113
## final  value 4.145113
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 83.575828
## iter  20 value 37.662203
## iter  30 value 29.702173
## final  value 29.662488
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 69.889060
## iter  20 value 27.055313
## iter  30 value 22.548207
```

```
## iter  40 value 16.114412
## iter  50 value 15.293513
## iter  60 value 14.002366
## iter  70 value 13.705712
## iter  80 value 13.493359
## iter  90 value 13.373253
## iter 100 value 13.187394
## final  value 13.187394
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 75.633160
## iter  20 value 14.658248
## iter  30 value 10.881707
## iter  40 value 9.961925
## iter  50 value 7.422724
## iter  60 value 4.378229
## iter  70 value 1.541240
## iter  80 value 0.802046
## iter  90 value 0.291719
## iter 100 value 0.209907
## final  value 0.209907
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 82.240693
## iter  20 value 40.748091
## iter  30 value 36.269941
## iter  40 value 36.260336
## final  value 36.260335
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 75.640291
## iter  20 value 15.180761
## iter  30 value 12.569415
## iter  40 value 11.009461
## iter  50 value 10.218926
## iter  60 value 9.548641
## iter  70 value 9.133008
## iter  80 value 8.798188
## iter  90 value 8.588481
## iter 100 value 8.323057
## final  value 8.323057
## stopped after 100 iterations
## # weights:  32 (31 variable)
```

```
## initial  value 314.688820
## iter  10 value 95.600106
## iter  20 value 20.692494
## iter  30 value 16.534027
## iter  40 value 13.324971
## iter  50 value 9.141687
## iter  60 value 3.882007
## iter  70 value 1.746874
## iter  80 value 0.741270
## iter  90 value 0.532091
## iter 100 value 0.449947
## final  value 0.449947
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 99.174536
## iter  20 value 37.193664
## iter  30 value 31.694760
## iter  40 value 31.597145
## iter  50 value 31.587186
## final  value 31.587182
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 95.603888
## iter  20 value 21.154754
## iter  30 value 17.215031
## iter  40 value 14.748754
## iter  50 value 12.391615
## iter  60 value 11.357143
## iter  70 value 11.186017
## iter  80 value 10.809884
## iter  90 value 10.074134
## iter 100 value 9.539128
## final  value 9.539128
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 94.835077
## iter  20 value 7.730219
## iter  30 value 0.245939
## iter  40 value 0.000191
## iter  40 value 0.000095
## iter  40 value 0.000014
## final  value 0.000014
## converged
```

```
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 83.083214
## iter  20 value 30.374819
## iter  30 value 27.413347
## iter  40 value 27.400797
## final  value 27.400785
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 94.848386
## iter  20 value 8.385373
## iter  30 value 4.311709
## iter  40 value 3.676109
## iter  50 value 3.379477
## iter  60 value 3.104884
## iter  70 value 2.993927
## iter  80 value 2.879793
## iter  90 value 2.687872
## iter 100 value 2.589918
## final  value 2.589918
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 95.320319
## iter  20 value 25.674199
## iter  30 value 21.766854
## iter  40 value 20.436423
## iter  50 value 19.138223
## iter  60 value 18.396940
## iter  70 value 17.821822
## iter  80 value 17.650918
## iter  90 value 17.221931
## iter 100 value 16.804096
## final  value 16.804096
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 98.562784
## iter  20 value 41.330464
## iter  30 value 35.599227
## iter  40 value 35.587325
## final  value 35.587298
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
```

```
## iter  10 value 95.323734
## iter  20 value 25.829056
## iter  30 value 22.080009
## iter  40 value 20.789943
## iter  50 value 19.802266
## iter  60 value 19.347874
## iter  70 value 19.107579
## iter  80 value 19.021185
## iter  90 value 18.804579
## iter 100 value 18.629172
## final  value 18.629172
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 88.306544
## iter  20 value 24.434809
## iter  30 value 20.991250
## iter  40 value 18.543049
## iter  50 value 17.051632
## iter  60 value 15.707461
## iter  70 value 14.827836
## iter  80 value 13.458468
## iter  90 value 12.472510
## iter 100 value 11.033065
## final  value 11.033065
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 92.077616
## iter  20 value 39.179779
## iter  30 value 36.676120
## iter  40 value 36.669367
## final  value 36.669363
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 88.310559
## iter  20 value 24.715258
## iter  30 value 21.493141
## iter  40 value 19.398451
## iter  50 value 18.194839
## iter  60 value 17.000896
## iter  70 value 16.045331
## iter  80 value 15.723034
## iter  90 value 14.938675
## iter 100 value 14.346869
```

```
## final  value 14.346869
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 99.859995
## iter  20 value 19.099504
## iter  30 value 12.288493
## iter  40 value 6.975078
## iter  50 value 0.348710
## iter  60 value 0.000828
## final  value 0.000011
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 104.561689
## iter  20 value 41.714204
## iter  30 value 31.730785
## iter  40 value 31.587481
## final  value 31.584761
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 99.864889
## iter  20 value 19.377346
## iter  30 value 13.556152
## iter  40 value 11.332502
## iter  50 value 9.747640
## iter  60 value 8.858211
## iter  70 value 8.641327
## iter  80 value 8.365530
## iter  90 value 7.562750
## iter 100 value 7.113703
## final  value 7.113703
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 91.584449
## iter  20 value 18.156078
## iter  30 value 14.064338
## iter  40 value 10.710406
## iter  50 value 5.168193
## iter  60 value 0.124467
## iter  70 value 0.000952
## iter  80 value 0.000116
## iter  90 value 0.000105
## final  value 0.000095
```

```
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 98.170814
## iter  20 value 33.491374
## iter  30 value 28.300378
## iter  40 value 28.284632
## iter  50 value 28.283179
## final  value 28.283177
## converged
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 91.591381
## iter  20 value 18.412765
## iter  30 value 14.707974
## iter  40 value 11.841933
## iter  50 value 9.709076
## iter  60 value 9.132595
## iter  70 value 8.437848
## iter  80 value 8.138102
## iter  90 value 7.797865
## iter 100 value 7.682270
## final  value 7.682270
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 80.811594
## iter  20 value 34.860799
## iter  30 value 29.748344
## iter  40 value 25.649828
## iter  50 value 22.011505
## iter  60 value 20.252810
## iter  70 value 19.804234
## iter  80 value 19.196936
## iter  90 value 18.024184
## iter 100 value 17.369120
## final  value 17.369120
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 85.727630
## iter  20 value 47.288624
## iter  30 value 46.752991
## final  value 46.752883
## converged
## # weights:  32 (31 variable)
```

```
## initial  value 314.688820
## iter  10 value 80.817006
## iter  20 value 35.071674
## iter  30 value 30.190236
## iter  40 value 26.759937
## iter  50 value 24.120202
## iter  60 value 23.504181
## iter  70 value 23.320088
## iter  80 value 23.080367
## iter  90 value 22.686854
## iter 100 value 22.281803
## final  value 22.281803
## stopped after 100 iterations
## # weights:  32 (31 variable)
## initial  value 314.688820
## iter  10 value 105.157298
## iter  20 value 30.739868
## iter  30 value 25.259498
## iter  40 value 23.369192
## iter  50 value 22.040138
## iter  60 value 21.236861
## iter  70 value 20.859680
## iter  80 value 20.751122
## iter  90 value 20.550964
## iter 100 value 20.468391
## final  value 20.468391
## stopped after 100 iterations

names(fits) <- models
```

Now we assume we don't have the outcomes of the test data and apply the model to predict the outcome, after which we shall review which model returns the highest accuracy based on the actual outcome.

```
# Create a matrix of predictions for the test set
pred <- sapply(fits, function(object)
  predict(object, newdata = brca_test_set))

acc <- colMeans(pred == brca_test_set$y2)

model_result <- data.frame(METHOD = models, ACCURACY = acc)
model_result

##                   METHOD  ACCURACY
## glm                  glm 0.9391304
```

```
## lda                 lda 0.9739130
## naive_bayes naive_bayes 0.9391304
## svmLinear     svmLinear 0.9652174
## knn                 knn 0.9391304
## gamLoess       gamLoess 0.9304348
## multinom       multinom 0.9826087
## qda                 qda 0.9478261
## rf                   rf 0.9478261
## adaboost       adaboost 0.9478261
```

The model ensemble accuracy average is **95.1%**

```
model_result <- bind_rows(model_result, data_frame
                          (METHOD="ensemble average", ACCURACY = mean(acc)))
mean(acc)
```

```
## [1] 0.9513043
```

Now we build an ensemble prediction based by majority vote of the first 10 models. We obtain an accuracy of **96.5%**

```
# build an ensemble prediction by majority vote and compute the accuracy of the ensemble.
votes <- rowMeans(pred == "M")
y_hat <- ifelse(votes > 0.5, "M", "B")
mean(y_hat == brca_test_set$y2)
```

```
## [1] 0.9652174
```

```
model_result <- bind_rows(model_result, data_frame(
  METHOD="ensemble majority vote", ACCURACY = mean(y_hat == brca_test_set$y2)))
```

3. RESULT

Here is a list of all models with their individual accuracy

METHOD	ACCURACY
glm	0.9391304
lda	0.9739130
naive_bayes	0.9391304
svmLinear	0.9652174

METHOD	ACCURACY
knn	0.9391304
gamLoess	0.9304348
multinom	0.9826087
qda	0.9478261
rf	0.9478261
adaboost	0.9478261
ensemble average	0.9513043
ensemble majority vote	0.9652174

We have two models that perform better than the ensemble

```
ind <- acc > mean(y_hat == brca_test_set$y2)
models[ind]
```

```
## [1] "lda"      "multinom"
```

4. CONCLUSION

The main objective of the project was to come up with a model that will best predict the right outcome based on the features, in this case, accuracy of the model. We were able to conclude that the best two models are **Linear Discriminant Analysis (lda)** and **Multinomial Log-linear (multinom)** with accuracy **97.4%** and **98.3%** respectively. Can we further tune parameters of the two models and evaluate which returns a more accurate prediction? Research worth exploring.