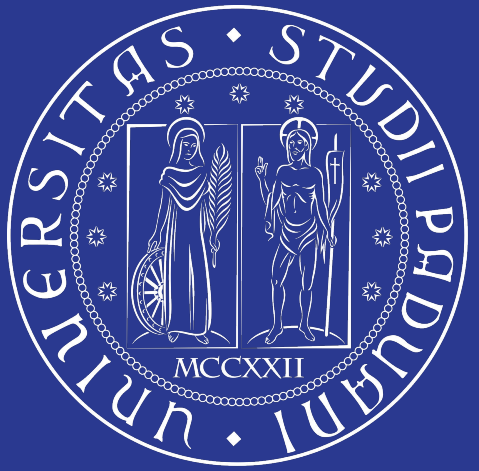
Breast Cancer Classification

Statistical Learning Project

Alessia d'Addario Mat. 2086506
Alice Ronzoni Mat 2076675



Problem Presentation

DATASET	Breast Tumor from Wisconsin University

GOAL	Find a model that correctly classifies breast tumor cancer given some features

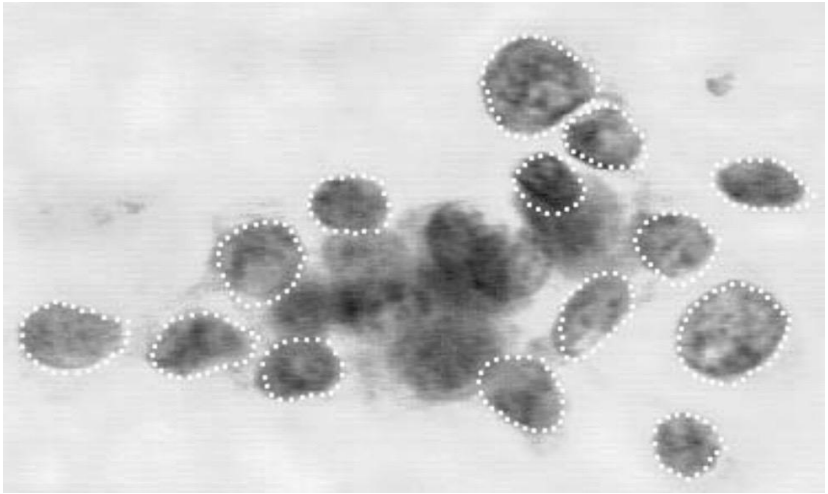
Features

ID
Diagnosis

Radius
Texture
Perimeter
Area
Smoothness
Compactness
Concavity
Concave_pts
Symmetry
Fractal_dim

Repeated for:

- mean
- SE
- worst



Clean and Filter Data

Check for missing values

```
sum(is.na(wdbc))
```

```
## [1] 0
```

Check for duplicate ID

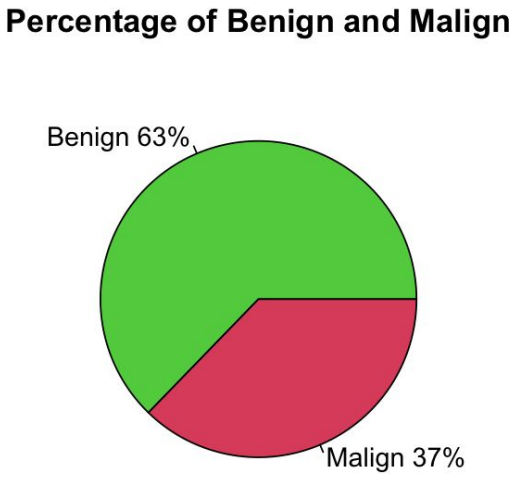
```
sum(duplicated(wdbc$id))
```

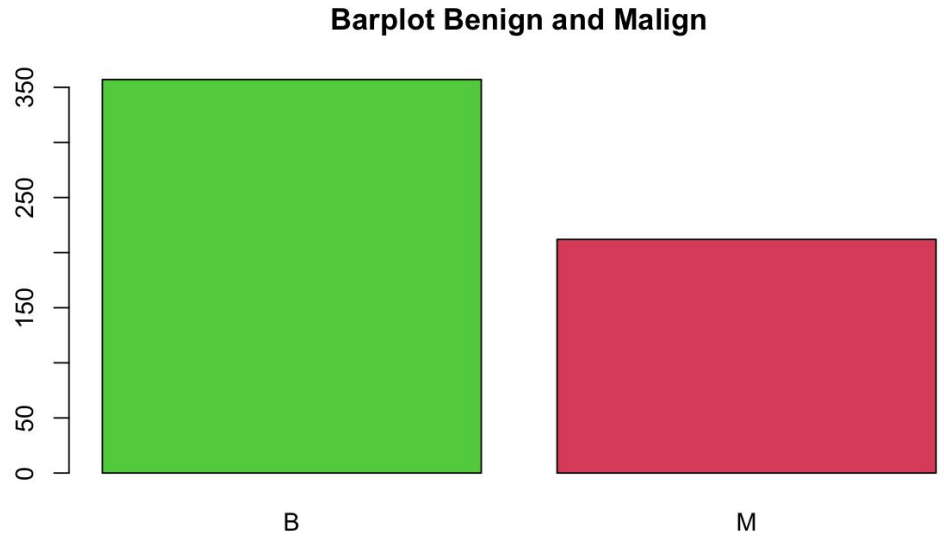
```
## [1] 0
```

Rename the columns

```
colnames(wdbc) = c('id', 'diagnosis',
                   'radius_mean',
                   'texture_mean',
                   'perimeter_mean',
                   'area_mean',
                   'smoothness_mean',...)
```

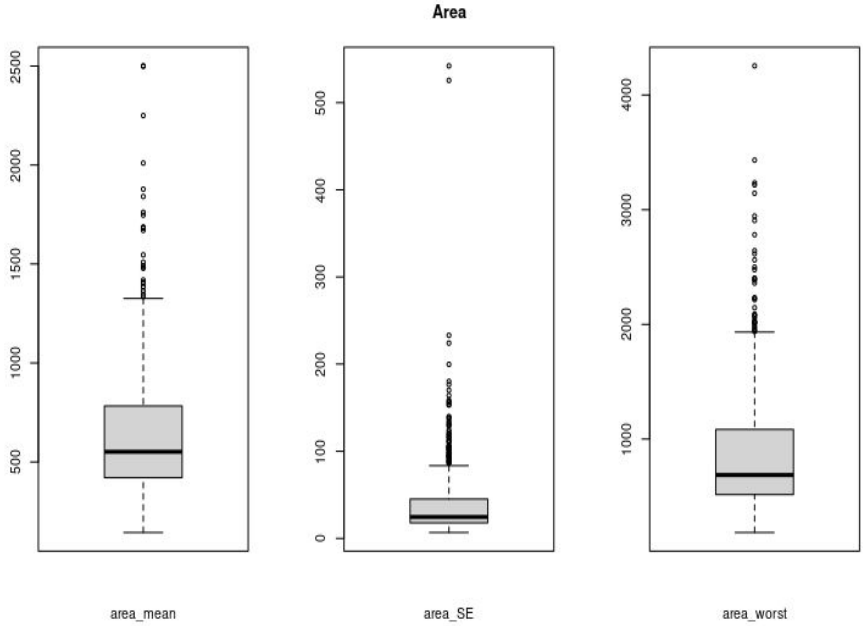
Data Exploration

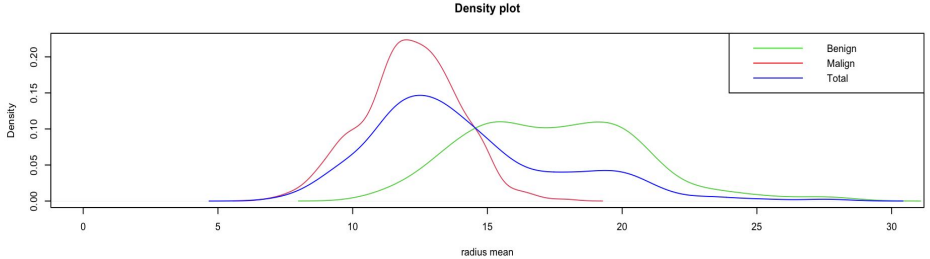


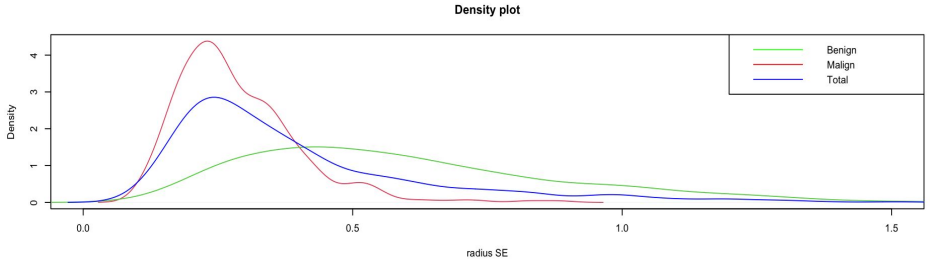


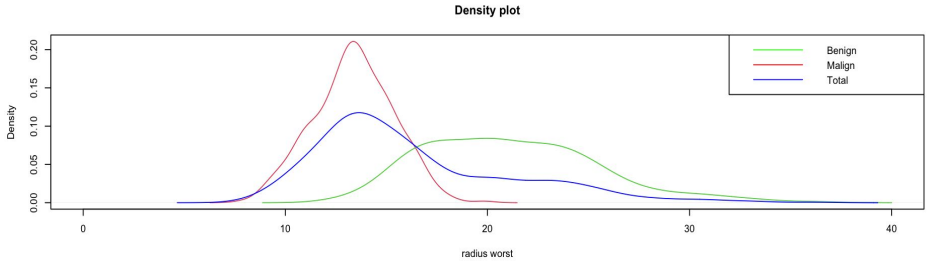
Data Exploration

We started by observing the plot produced by our features.
Here we report the ones obtain from the Area and Concavity values:



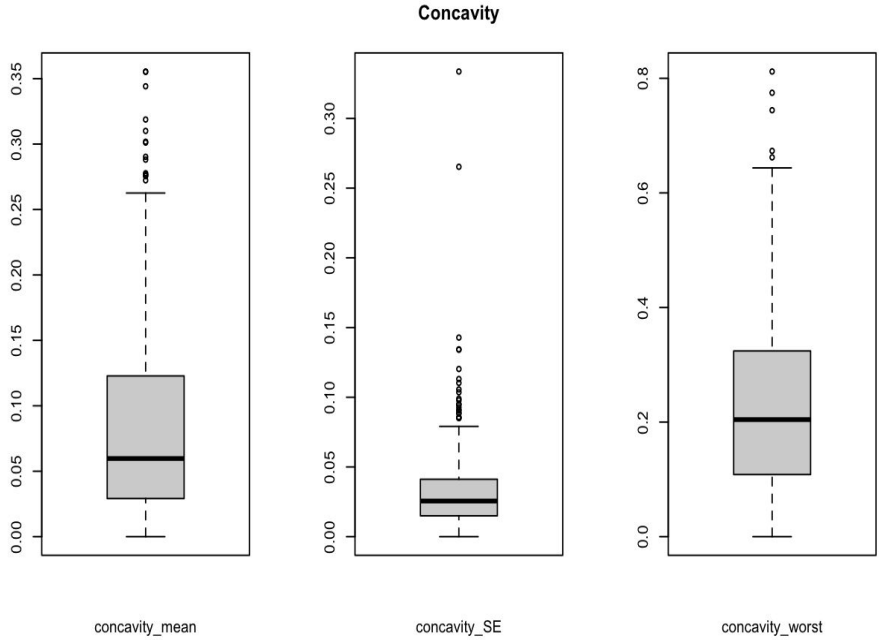


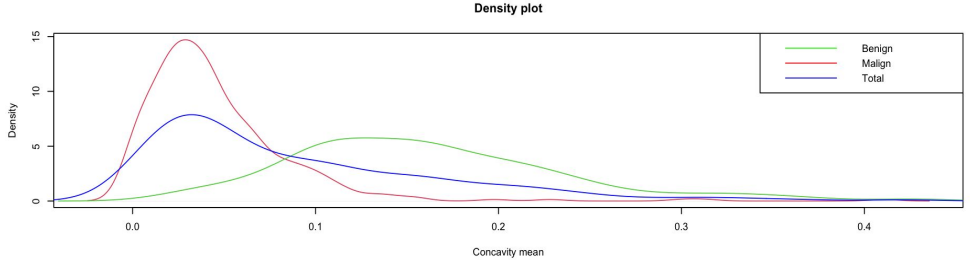


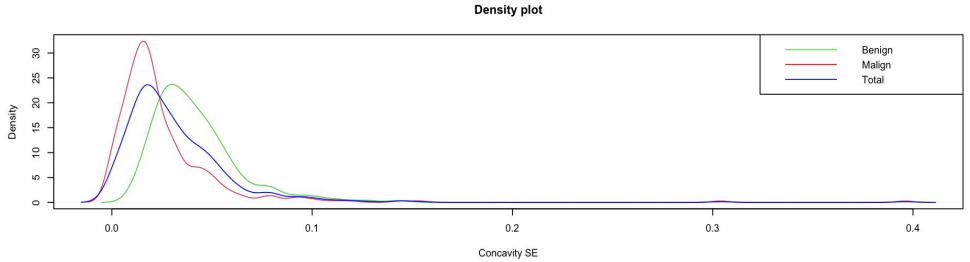


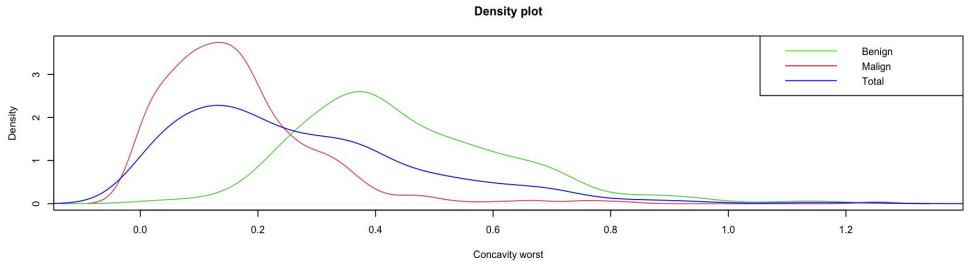
Data Exploration

Since the density plots don't resemble a very good bell shape, before further explorations we checked for the normality of the data.

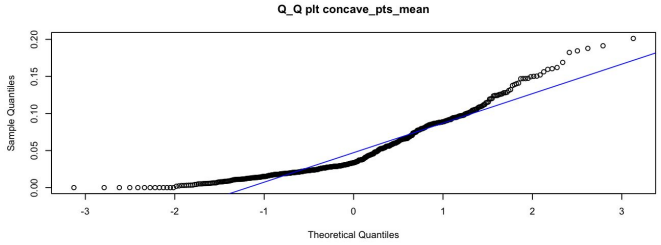


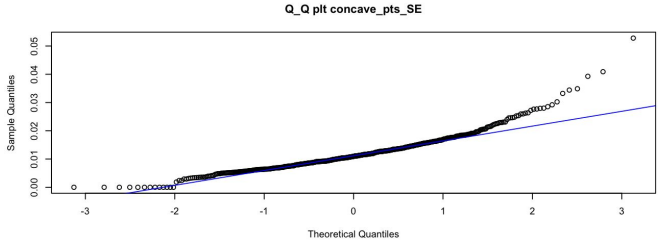


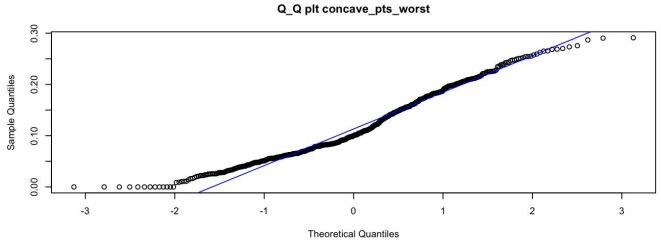


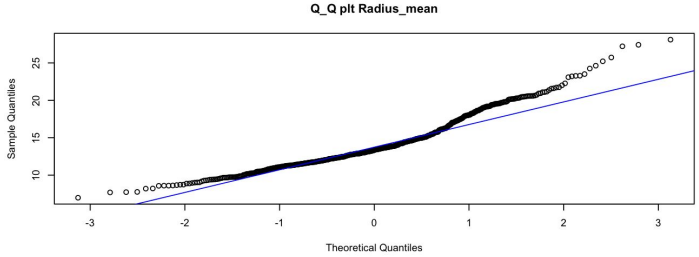


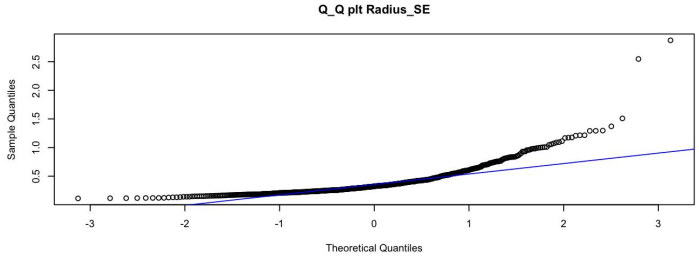
Data Exploration

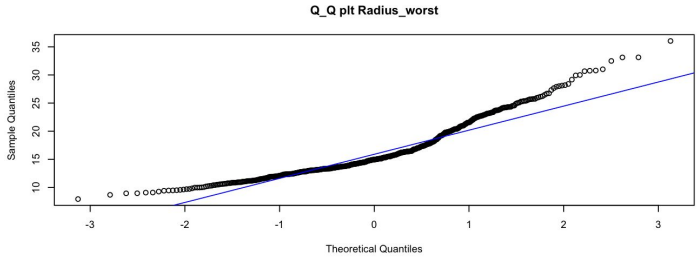








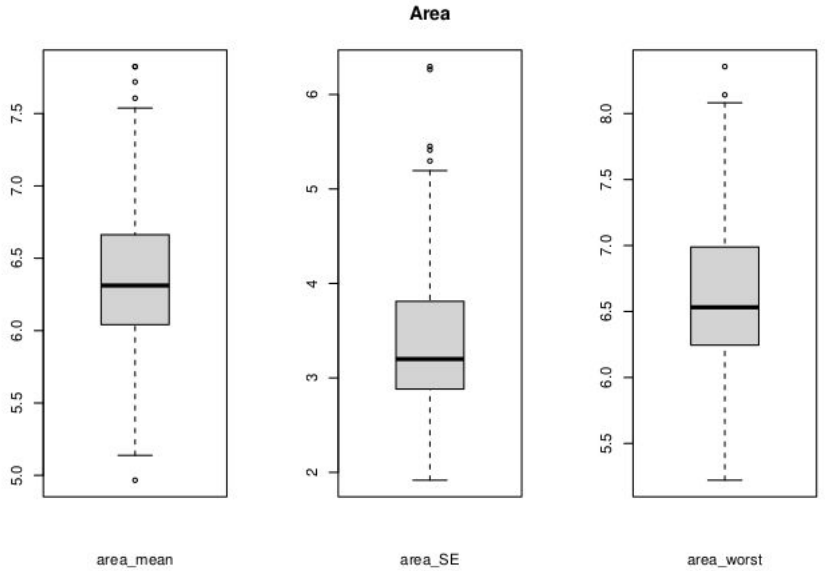


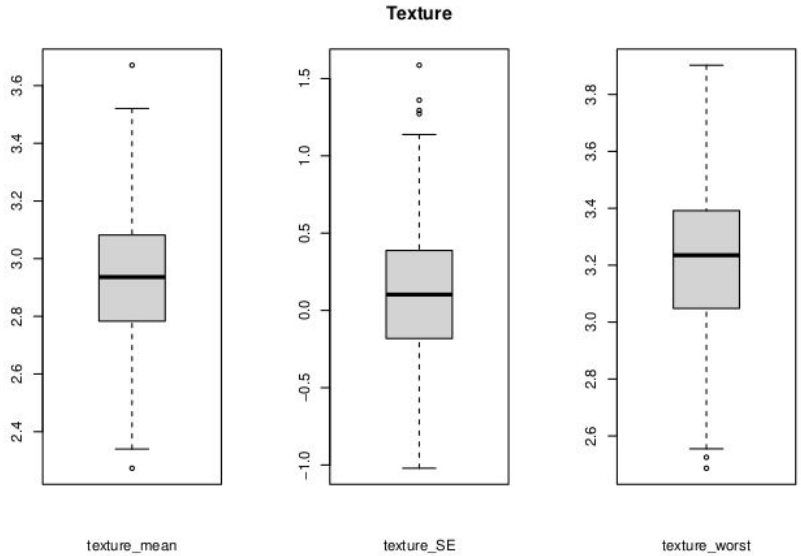


As supposed our features lack of normality behaviour

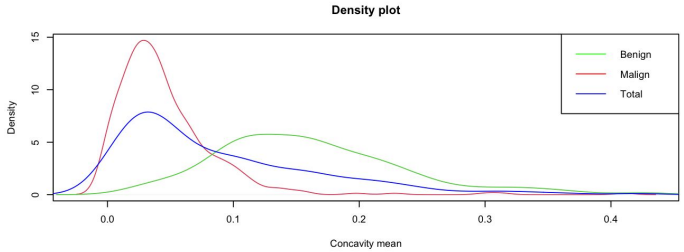
For this reason we decided to apply a non linear transformation to the predictor :

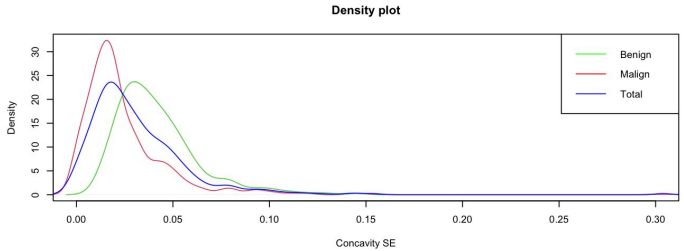
log()

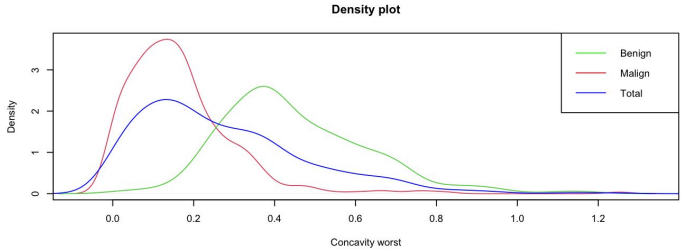

Area
area_mean
area_SE
area_worst

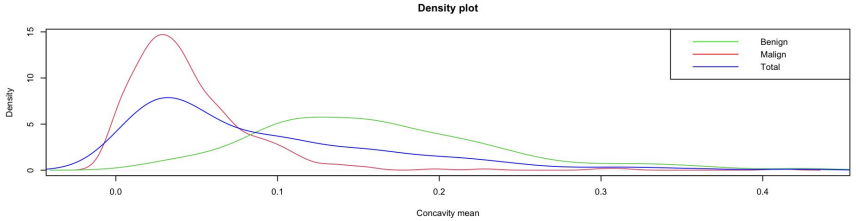

Texture
texture_mean
texture_SE
texture_worst

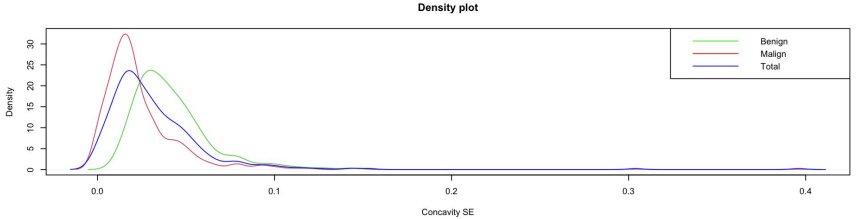
Data Exploration cnt

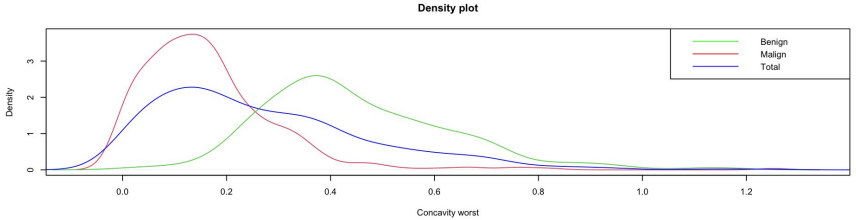


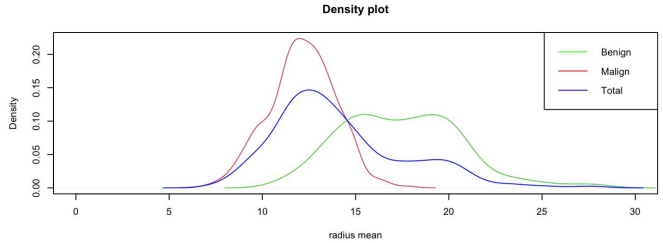


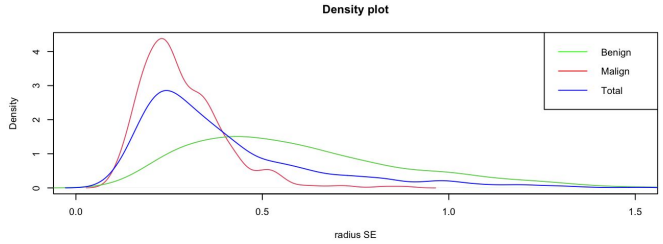


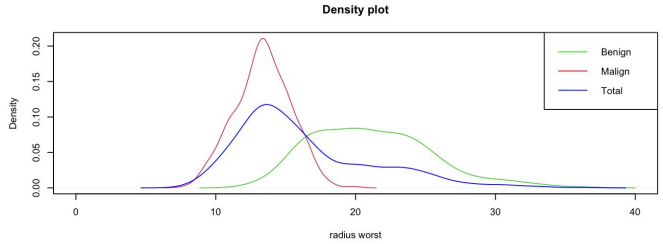




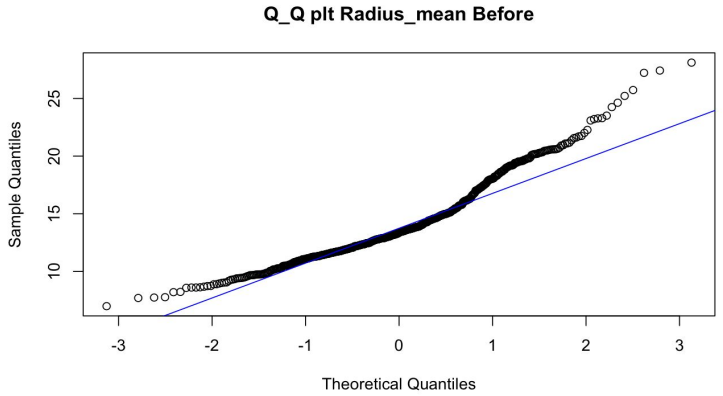


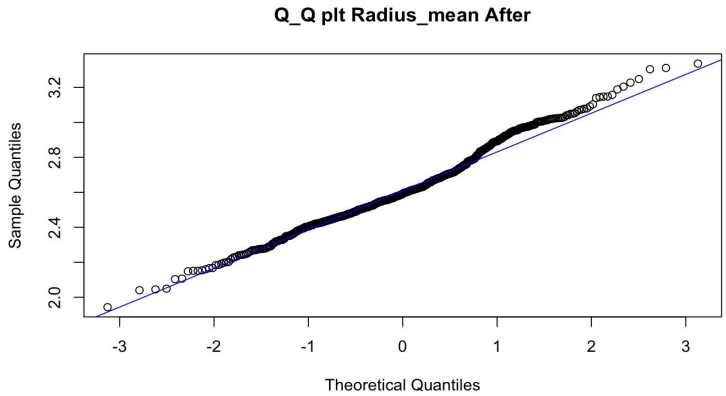

Density plot
Benign
Malign
Total
Density
0.00
0.05
0.10
0.15
0.20
0
5
10
15
20
25
30
radius mean

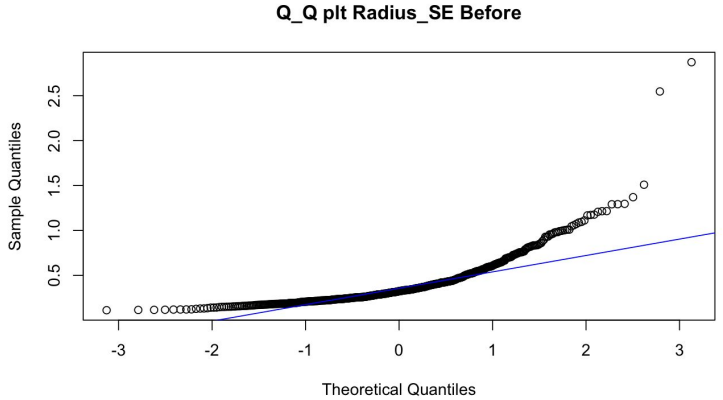

Density plot
Benign
Malign
Total
Density
0
1
2
3
4
0.0
0.5
1.0
1.5
radius SE

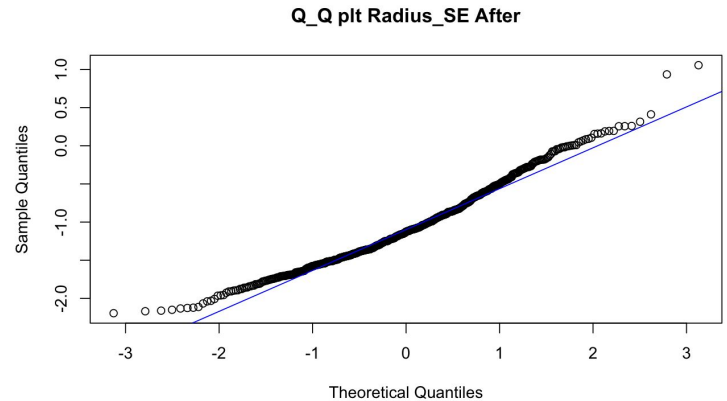

Density plot
Benign
Malign
Total
Density
0.00
0.05
0.10
0.15
0.20
0
10
20
30
40
radius worst

Let’s see the outcome of the transformation :

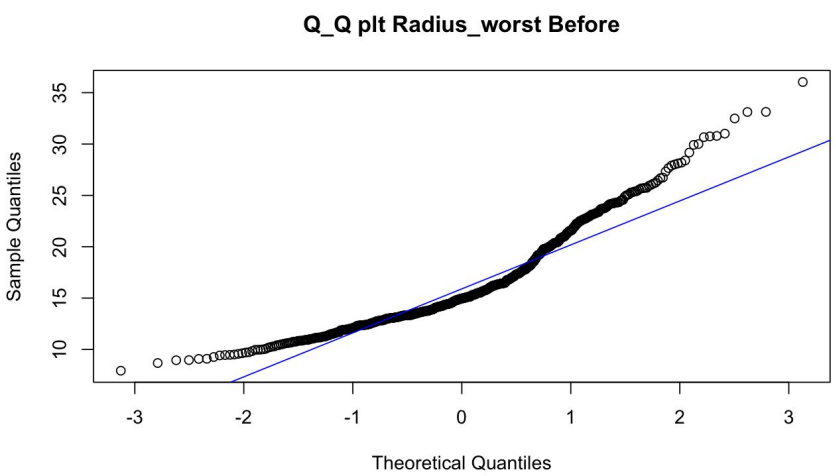


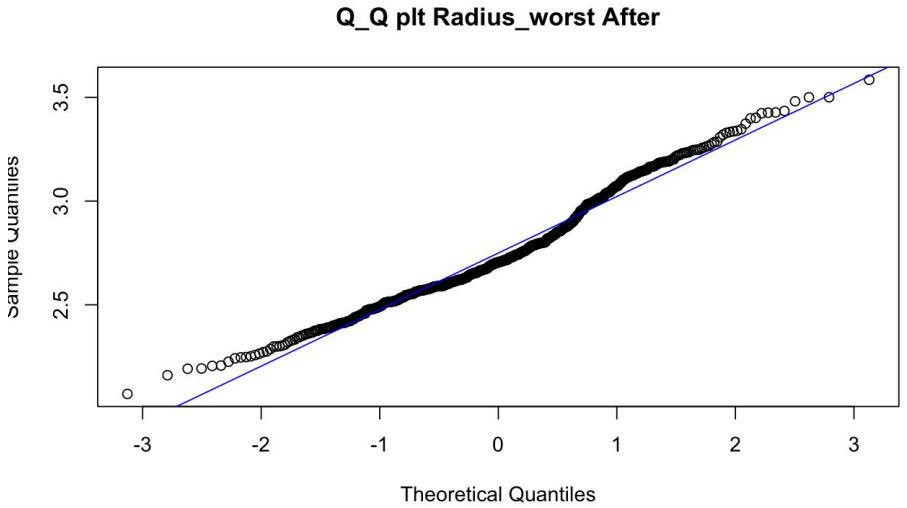






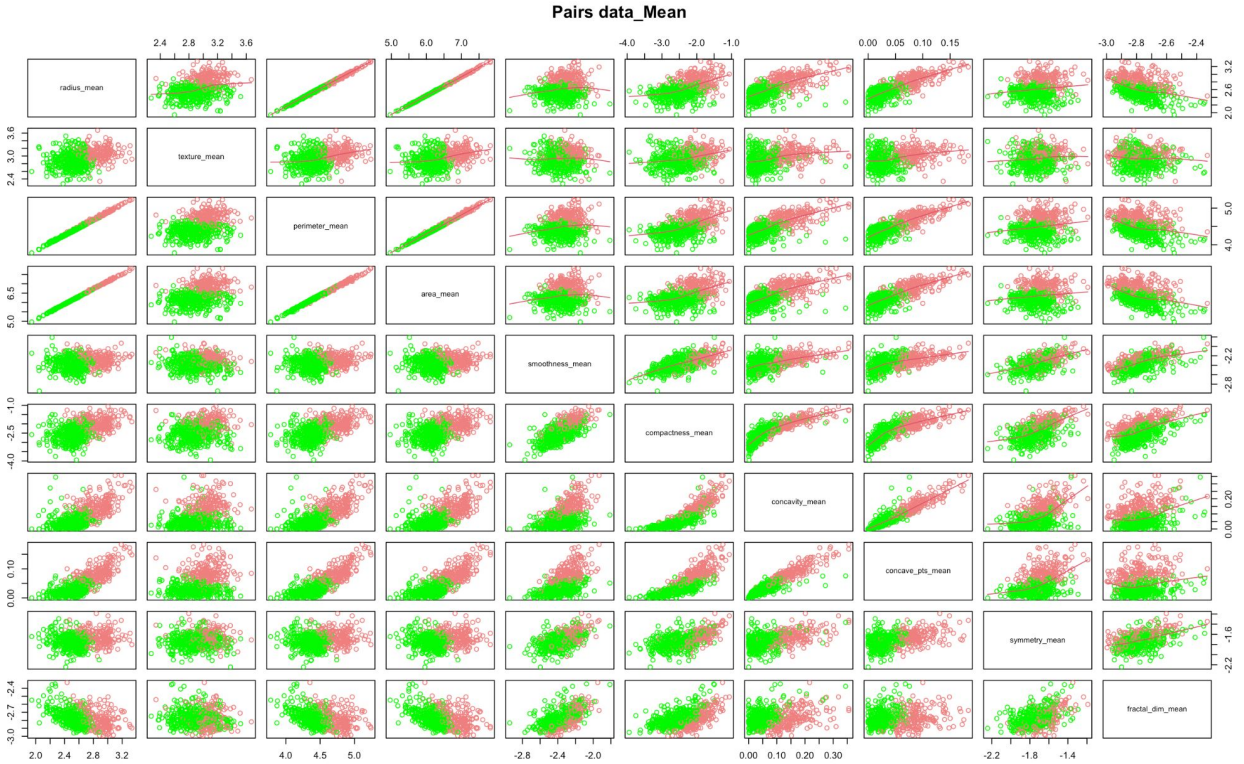
Trova un titolo sta di fatto che sono già a confronto





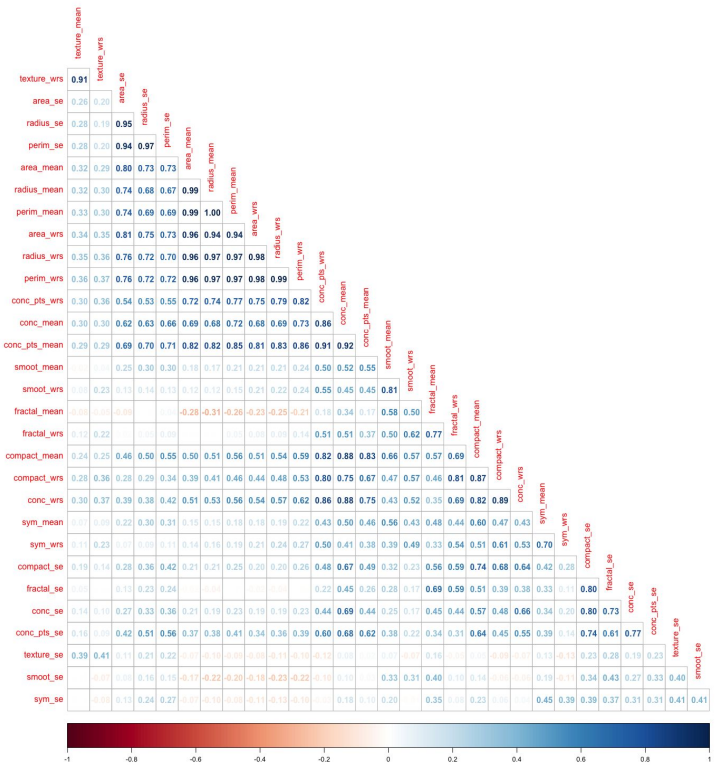
Pairs Plot

To visualize a pairwise comparison of the correlations between our variables we plotter the Pair Plot.
For simplicity we consider attributes referring to the same group.
Here we report the comparison between the mean values.



Correlation Matrix

And with the help of the covariance matrix, we were able to have numerical values of such correlation.



Problem

We noticed some high correlation values that could hide a collinearity problem.
We will need to take it in consideration during the model analysis.

Classification Models

Train and Test Set

We decided to divide the dataset into two smaller subsets:

- Train set (80%)
- Test set (20%)

```
set.seed(161)
sample <- sample(1:569, size=455, replace= FALSE)
wdbc_train <- wdbc_df[sample,] #training set
wdbc_test <- wdbc_df[-sample,] #test set
```

Logistic Regression as Classification

As first Model we used Logistic Regression.

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

To begin we tried using all the parameters, but we obtained two warning messages.
This lead us to look at the VIF of our model:

```
vif(lr_model_0)
```

```
##       radius_mean      texture_mean    perimeter_mean         area_mean
##       15218.44015         167.24240       43290.64843       21723.85577
##   smoothness_mean  compactness_mean     concavity_mean  concave_pts_mean
##        141.25014         179.55738         184.72538         201.90282
##    symmetry_mean  fractal_dim_mean         radius_SE        texture_SE
##         31.32979          55.41660        3202.03068          57.24564
##     perimeter_SE          area_SE     smoothness_SE    compactness_SE
##        422.93630        4076.49099          93.74082         169.58219
##     concavity_SE     concave_pts_SE       symmetry_SE    fractal_dim_SE
##        205.16199         127.87768         143.77884         274.01617
##     radius_worst     texture_worst   perimeter_worst        area_worst
##       4943.09284         276.74398        1223.58974        6382.22674
## smoothness_worst compactness_worst   concavity_worst concave_pts_worst
##        200.64867         427.58625         118.53194         126.95973
##   symmetry_worst fractal_dim_worst
##         72.79374         440.19752
```

Logistic Regression as Classification

Result of the VIF function after applying it different times

```
##      texture_mean   smoothness_mean  concave_pts_mean     symmetry_mean
##          6.772479         4.353268         4.337390         3.119615
##         radius_SE    smoothness_SE      concavity_SE       symmetry_SE
##          2.502346         3.163113         2.173610         2.559119
##    fractal_dim_SE     radius_worst    texture_worst concave_pts_worst
##          3.307410         3.098307         6.523176         4.013336
```

Result of the Backward selection

Starting with the model obtained after the application of the VIF method, we use the Backward Selection.
At the end our model contains 7 features:

- *smoothness_mean*
- *radius_SE*
- *concavity_SE*
- *fractal_dim_SE*
- *radius_worst*
- *texture_worst*
- *concave_pts_worst*

```
## glm(formula = diagnosis ~ . - area_worst - radius_mean - perimeter_mean -
##     perimeter_worst - area_SE - concavity_mean - area_mean -
##     fractal_dim_worst - compactness_worst - concavity_worst -
##     texture_SE - perimeter_SE - fractal_dim_mean - concave_pts_SE -
##     symmetry_worst - compactness_mean - smoothness_worst - compactness_SE -
##     smoothness_SE - concave_pts_mean - symmetry_SE - texture_mean -
##     symmetry_mean, family = binomial, data = wdbc_train)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -98.633     19.772  -4.989 6.08e-07 ***
## smoothness_mean      9.481      4.586   2.067 0.038720 *
## radius_SE            3.281      1.583   2.073 0.038210 *
## concavity_SE        38.238     19.172   1.994 0.046107 *
## fractal_dim_SE      -3.119      1.289  -2.420 0.015540 *
## radius_worst        21.034      5.368   3.919 8.90e-05 ***
## texture_worst       11.860      2.572   4.612 4.00e-06 ***
## concave_pts_worst   63.288     19.081   3.317 0.000911 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 600.338  on 454  degrees of freedom
## Residual deviance:  54.248  on 447  degrees of freedom
## AIC: 70.248
##
## Number of Fisher Scoring iterations: 10
```

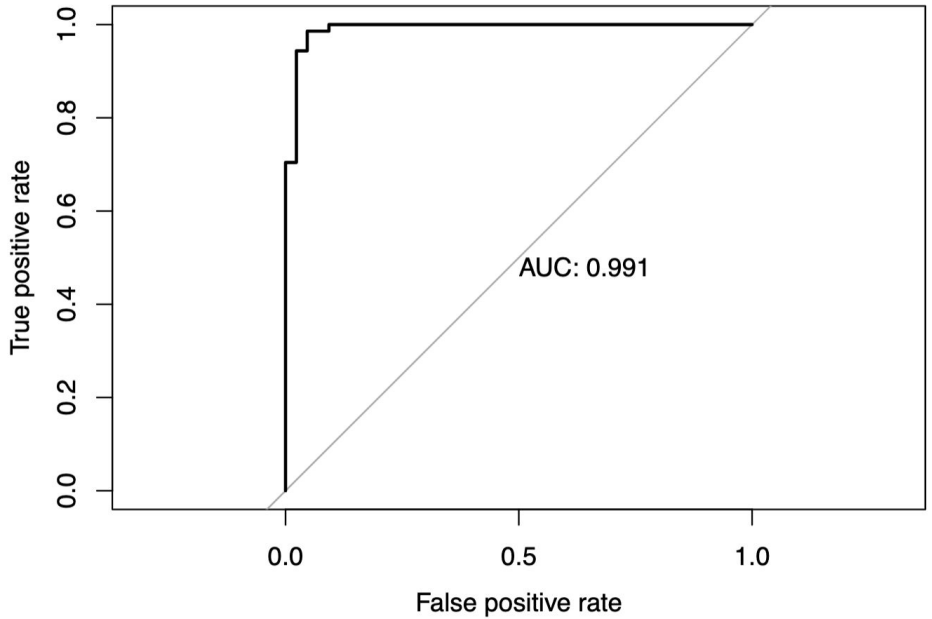
Confusion Matrix on Logistic Regression

```
##
## logistic_predictions   B   M Sum
##                    B  70   2  72
##                    M   1  41  42
##                    Sum 71  43 114
```

Error	0.02631579
Specificity	0.9534884
Sensitivity	0.9859155
Precision	0.9722222

ROC Curve

We plotted the Roc curve that simultaneously display the True and False positive rates for all possible threshold

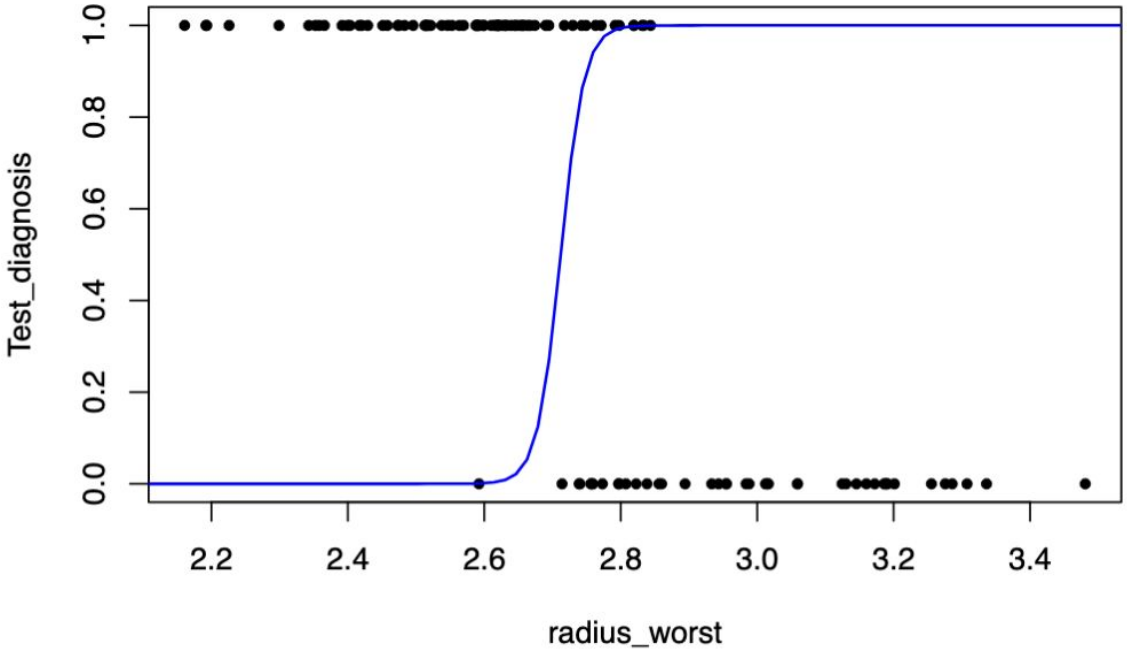


and we used the best one to do predictions

```
result <- coords(roc.out23, x = "best")
result
```

```
##   threshold specificity sensitivity
## 1 0.4588467   0.9534884   0.9859155
```

Logistic plot

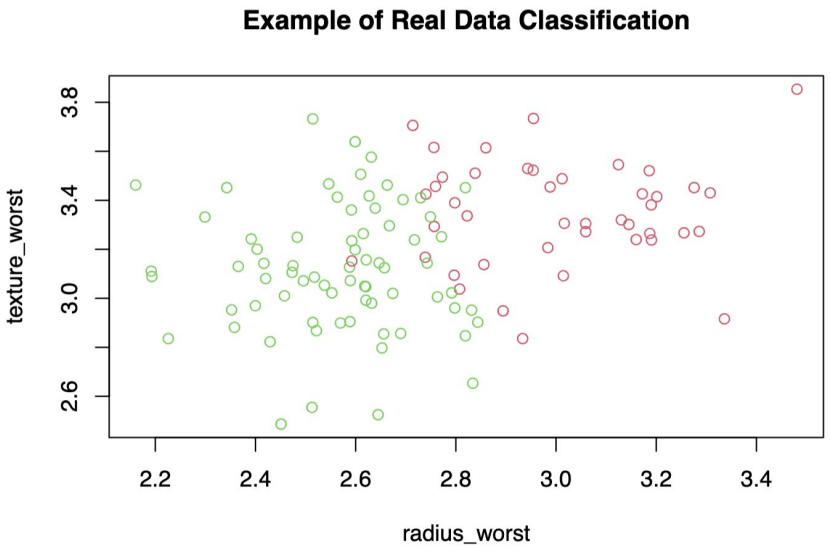


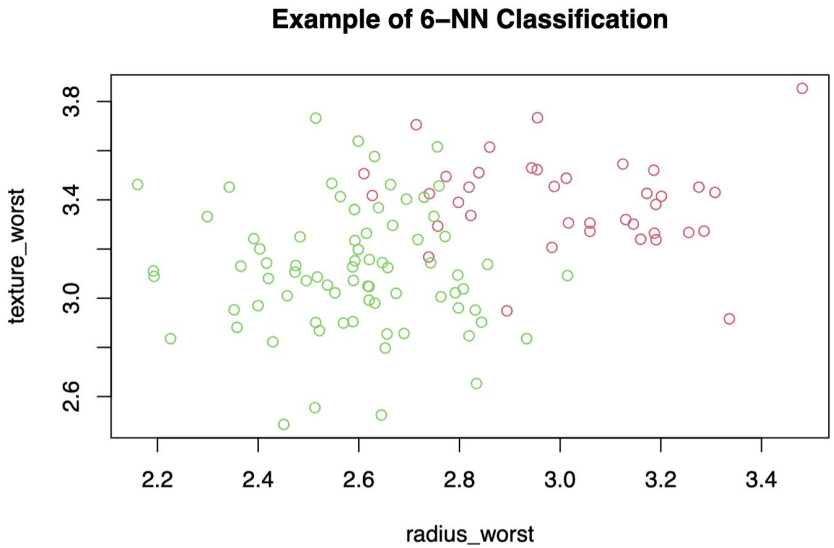
KNN Model

```
##
## knn_predictor   B   M Sum
##             B  68   8  76
##             M   3  35  38
##             Sum 71  43 114
```

Error	0.09649123
Specificity	0.8139535
Sensitivity	0.9577465
Precision	0.8947368

Before and After applying KNN





Bayes Classifier

```
##          predictions
##              B    M Sum
##    B        69    2  71
##    M         3   40  43
##    Sum      72   42 114
```

Error	0.06140351
Specificity	0.952381
Sensitivity	0.9583333
Precision	0.971831

Linear Discriminant Analysis

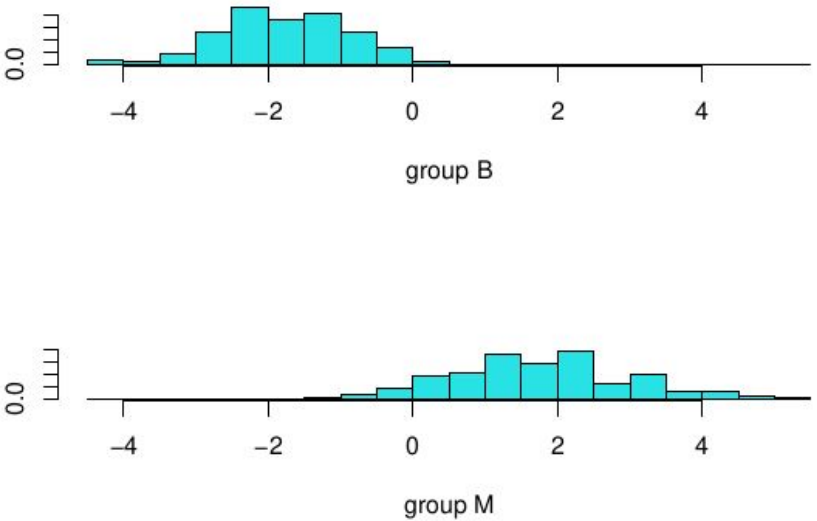
```
## Call:
## lda(diagnosis ~ . - area_worst - radius_mean - perimeter_mean -
##     perimeter_worst - area_SE - concavity_mean - area_mean -
##     fractal_dim_worst - compactness_worst - concavity_worst -
##     texture_SE - perimeter_SE - fractal_dim_mean - concave_pts_SE -
##     symmetry_worst - compactness_mean - smoothness_worst - compactness_SE,
##     data = wdbc_train)
##
## Prior probabilities of groups:
##         B         M
## 0.6285714 0.3714286
##
## Group means:
##   texture_mean smoothness_mean concave_pts_mean symmetry_mean  radius_SE
## B     2.861709       -2.392849       0.02521844     -1.762361 -1.3343108
## M     3.063533       -2.284704       0.08404616     -1.657907 -0.6263332
##   smoothness_SE concavity_SE symmetry_SE fractal_dim_SE radius_worst
## B     -5.022221   0.02468876   -3.941404      -5.816084     2.585837
## M     -5.049640   0.04052123   -3.986387      -5.636562     3.038655
##   texture_worst concave_pts_worst
## B      3.134965        0.07144043
## M      3.364480        0.16707627
##
## Coefficients of linear discriminants:
##                          LD1
## texture_mean       0.34535767
## smoothness_mean   -0.16054822
## concave_pts_mean  -0.64741100
## symmetry_mean      1.10080941
## radius_SE          0.57397426
## smoothness_SE      0.45644163
## concavity_SE      -6.95341580
## symmetry_SE       -0.01774347
## fractal_dim_SE    -0.13822009
## radius_worst       2.65811356
## texture_worst      1.32828108
## concave_pts_worst 17.10773724
```

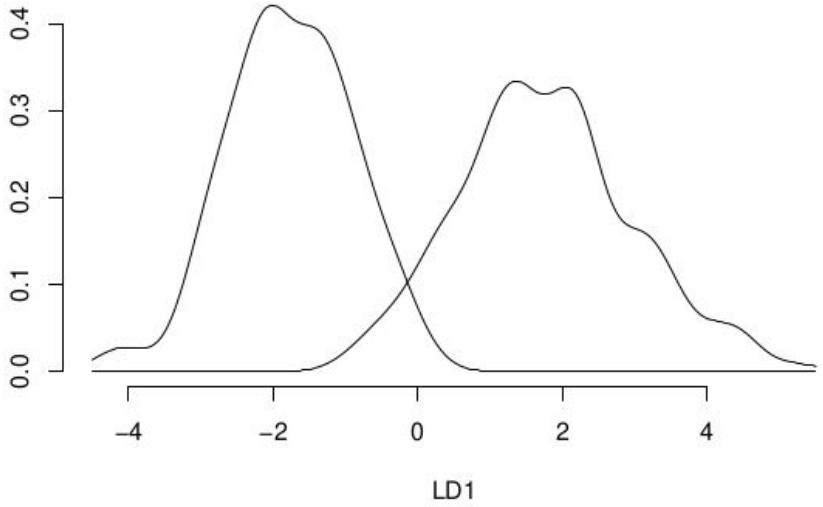
Linear Discriminant Analysis

```
##
## lda_class   B   M Sum
##         B  70   4  74
##         M   1  39  40
##         Sum 71  43 114
```

Error	0.04385965
Specificity	0.9069767
Sensitivity	0.9859155
Precision	0.9459459

Linear Discriminant Analysis





Quadratic Discriminant Analysis

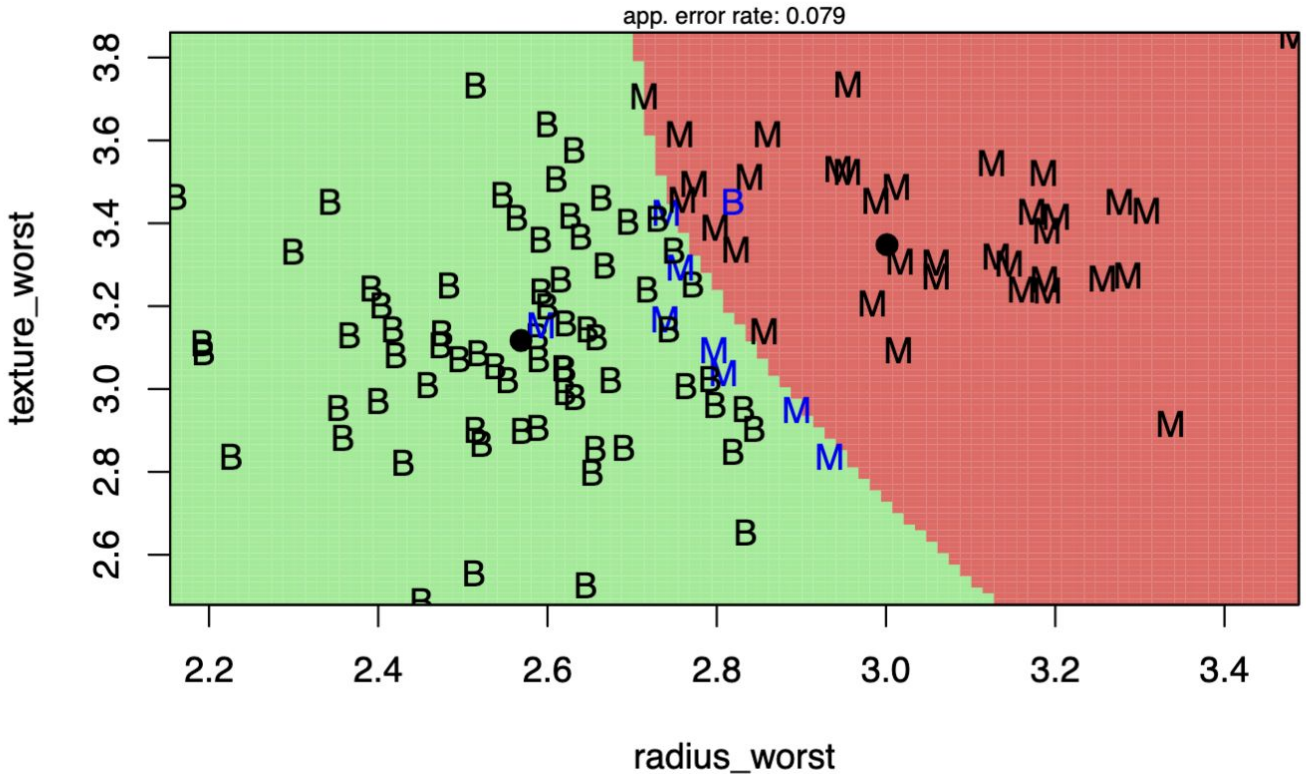
```
## Call:
## qda(diagnosis ~ . - area_worst - radius_mean - perimeter_mean -
##     perimeter_worst - area_SE - concavity_mean - area_mean -
##     fractal_dim_worst - compactness_worst - concavity_worst -
##     texture_SE - perimeter_SE - fractal_dim_mean - concave_pts_SE -
##     symmetry_worst - compactness_mean - smoothness_worst - compactness_SE,
##     data = wdbc_train)
##
## Prior probabilities of groups:
##         B         M
## 0.6285714 0.3714286
##
## Group means:
##   texture_mean smoothness_mean concave_pts_mean symmetry_mean  radius_SE
## B     2.861709       -2.392849       0.02521844     -1.762361 -1.3343108
## M     3.063533       -2.284704       0.08404616     -1.657907 -0.6263332
##   smoothness_SE concavity_SE symmetry_SE fractal_dim_SE radius_worst
## B     -5.022221   0.02468876   -3.941404      -5.816084     2.585837
## M     -5.049640   0.04052123   -3.986387      -5.636562     3.038655
##   texture_worst concave_pts_worst
## B      3.134965        0.07144043
## M      3.364480        0.16707627
```

Quadratic Discriminant Analysis

```
##
## qda_class   B   M Sum
##         B  69   4  73
##         M   2  39  41
##         Sum 71  43 114
```

Error	0.05263158
Specificity	0.9069767
Sensitivity	0.971831
Precision	0.9452055

QDA plot


app. error rate: 0.079
texture_worst
radius_worst
2.6
2.8
3.0
3.2
3.4
3.6
3.8
2.2
2.4
2.6
2.8
3.0
3.2
3.4

Final Considerations

	LOGISTIC REGRESSION	KNN	BAYES CLASSIFIER	LDA	QDA
ERROR	0.02631579	0.09649123	0.06140351	0.04385965	0.05263158
SPECIFICITY	0.9534884	0.8139535	0.952381	0.9069767	0.9069767
SENSITIVITY	0.9859155	0.9577465	0.9583333	0.9859155	0.971831
PRECISION	0.9722222	0.8947368	0.971831	0.9459459	0.9452055