Medical diagnostics: Breast Cancer Wisconsin (Diagnostic) Project

Patrycja Cieślachowska,
Aleksandra Winiarska

Data Mining
dr Adam Zagdański

December 3, 2023

Contents

1 Introduction

In this paper, we will analyze a dataset on the diagnosis of breast cancer. In the project, we will consider which factors available in the dataset, well describe the types of cancer: malignant and benign.

Medics describe a benign tumor as a cancer that has distinct, smooth, regular borders. A malignant tumor has irregular borders and grows faster than a benign tumor. A malignant tumor can also spread to other parts of your body. A benign tumor can become quite large, but it will not invade nearby tissue or spread to other parts of your body. [1]

As a result, we would like to create a model that classifies types of cancer effectively. Faster accurate diagnosis allows choosing the right treatment, allowing the patient to enjoy further life.

1.1 Data description

Analyzed dataset is entitled Breast Cancer Wisconsin (Diagnostic) and can be downloaded at archive.ics.uci.edu. The data contains 569 instances, 30 features used for diagnosting a patient (the mean, standard error, and ”worst” or largest – mean of the three largest values – of these features were computed for each image) and a columns on diagnosis and patient ID. They have no missing or unusual values.

Ten real-valued features are computed for each cell nucleus. The unique names of columns in the dataset are:

1. ID number,
2. diagnosis (M – malignant, B – benign),
3. radius (mean of distances from center to points on the perimeter),
4. texture (standard deviation of gray-scale values),
5. perimeter,
6. area,
7. smoothness (local variation in radius lengths),
8. compactness ($\text{perimeter}^2/\text{area} - 1.0$),
9. concavity (severity of concave portions of the contour),
10. concave points (number of concave portions of the contour),
11. symmetry,
12. fractal dimension (”coastline approximation” -1).

All feature values are recoded with four significant digits.

2 Methods

2.1 Exploratory Data Analysis

To perform Exploratory Data Analysis we used libraries: DataExplorer, scales, stats, utils, graphics, grDevices, ggplot2 and base. All main functions are listed below:

- is.factor(),
- is.na(),
- min(),
- max(),
- quantile(),
- median(),
- mean(),
- var(),
- sd(),
- IQR(),
- round(),
- apply(),
- sapply(),
- shapiro.test(),
- introduce(),
- plot_intro(),
- par(),
- plot_histogram(),
- plot_boxplot(),
- plot_density(),
- plot_qq(),
- barplot(),
- plot_correlation(),
- pairs.

2.2 Classification

To perform Classification we used libraries: caret, klaR, ipred, rpart and randomForest. All main functions are listed below:

- findCorrelation(),
- cor(),
- sample(),
- prop.table(),
- table(),
- stepclass(),
- lda(),
- qda(),
- ipredknn(),
- glm(),
- rpart(),
- bagging(),
- randomForest(),
- predict(),
- errorest().

3 Exploratory Data Analysis

3.1 Summary of the data

As can be seen in the table below, the data consists of 569 rows and 32 columns, including 1 column with discrete data (diagnosis) and 31 columns with continuous data. There are 18208 total observations, and the data does not contain any missing values or columns.

```
##                              [,1]
## 1                  rows     569
## 2               columns      32
## 3      discrete_columns       1
## 4     continous_columns      31
## 5   all_missing_columns      0
## 6 total_missing_values       0
## 7         complete_rows    569
## 8    total_observations  18208
## 9          memory_usage 148984
```

We can see the problem of class imbalance in the data – we have a total of 357 patients with benign cancer and 212 with malignant cancer.

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

In the table below, we present the basic summary of data values from each column with continuous values. All values have been rounded to two decimal places. Note that many variables have a variance close to 0. The columns describing area, perimeter and texture take the largest average values (no matter if these columns refer to the mean value, standard error or worst value). The vast majority of features are characterized by a small interquartile range, indicating little variability in the trait. For the variables Perimeter1, Perimeter3, and Area1, Area2, and Area3, these values are significantly larger.

```
##                       Min     Q1 Median   Mean      Q3     Max       Var     Sd
## Radius1              6.98  11.70  13.37  14.13   15.78   28.11     12.42   3.52
## Texture1             9.71  16.17  18.84  19.29   21.80   39.28     18.50   4.30
## Perimeter1          43.79  75.17  86.24  91.97  104.10  188.50    590.44  24.30
## Area1              143.50 420.30 551.10 654.89  782.70 2501.00 123843.55 351.91
## Smoothness1          0.05   0.09   0.10   0.10    0.11    0.16      0.00   0.01
## Compactness1         0.02   0.06   0.09   0.10    0.13    0.35      0.00   0.05
## Concavity1           0.00   0.03   0.06   0.09    0.13    0.43      0.01   0.08
## Concave_points1      0.00   0.02   0.03   0.05    0.07    0.20      0.00   0.04
## Symmetry1            0.11   0.16   0.18   0.18    0.20    0.30      0.00   0.03
## Fractal_dimension1   0.05   0.06   0.06   0.06    0.07    0.10      0.00   0.01
## Radius2              0.11   0.23   0.32   0.41    0.48    2.87      0.08   0.28
## Texture2             0.36   0.83   1.11   1.22    1.47    4.88      0.30   0.55
## Perimeter2           0.76   1.61   2.29   2.87    3.36   21.98      4.09   2.02
## Area2                6.80  17.85  24.53  40.34   45.19  542.20   2069.43  45.49
## Smoothness2          0.00   0.01   0.01   0.01    0.01    0.03      0.00   0.00
## Compactness2         0.00   0.01   0.02   0.03    0.03    0.14      0.00   0.02
## Concavity2           0.00   0.02   0.03   0.03    0.04    0.40      0.00   0.03
## Concave_points2      0.00   0.01   0.01   0.01    0.01    0.05      0.00   0.01
## Symmetry2            0.01   0.02   0.02   0.02    0.02    0.08      0.00   0.01
## Fractal_dimension2   0.00   0.00   0.00   0.00    0.00    0.03      0.00   0.00
## Radius3              7.93  13.01  14.97  16.27   18.79   36.04     23.36   4.83
## Texture3            12.02  21.08  25.41  25.68   29.72   49.54     37.78   6.15
## Perimeter3          50.41  84.11  97.66 107.26  125.40  251.20   1129.13  33.60
## Area3              185.20 515.30 686.50 880.58 1084.00 4254.00 324167.39 569.36
## Smoothness3          0.07   0.12   0.13   0.13    0.15    0.22      0.00   0.02
## Compactness3         0.03   0.15   0.21   0.25    0.34    1.06      0.02   0.16
## Concavity3           0.00   0.11   0.23   0.27    0.38    1.25      0.04   0.21
## Concave_points3      0.00   0.06   0.10   0.11    0.16    0.29      0.00   0.07
## Symmetry3            0.16   0.25   0.28   0.29    0.32    0.66      0.00   0.06
## Fractal_dimension3   0.06   0.07   0.08   0.08    0.09    0.21      0.00   0.02
##                        IQR
## Radius1               4.08
## Texture1              5.63
## Perimeter1           28.93
## Area1               362.40
## Smoothness1           0.02
## Compactness1          0.07
## Concavity1            0.10
## Concave_points1       0.05
## Symmetry1             0.03
## Fractal_dimension1    0.01
## Radius2               0.25
## Texture2              0.64
## Perimeter2            1.75
## Area2                27.34
## Smoothness2           0.00
## Compactness2          0.02
## Concavity2            0.03
## Concave_points2       0.01
## Symmetry2             0.01
## Fractal_dimension2    0.00
## Radius3               5.78
## Texture3              8.64
## Perimeter3           41.29
## Area3               568.70
## Smoothness3           0.03
## Compactness3          0.19
## Concavity3            0.27
## Concave_points3       0.10
## Symmetry3             0.07
## Fractal_dimension3    0.02
```

3.2 Histogram of qualitative feature

Figure 1 shows the histogram of the only qualitative feature in the dataset – Diagnosis. One may easily see that we are observing the previously described problem of class imbalance. Almost twice as many people described in the dataset have benign cancer.

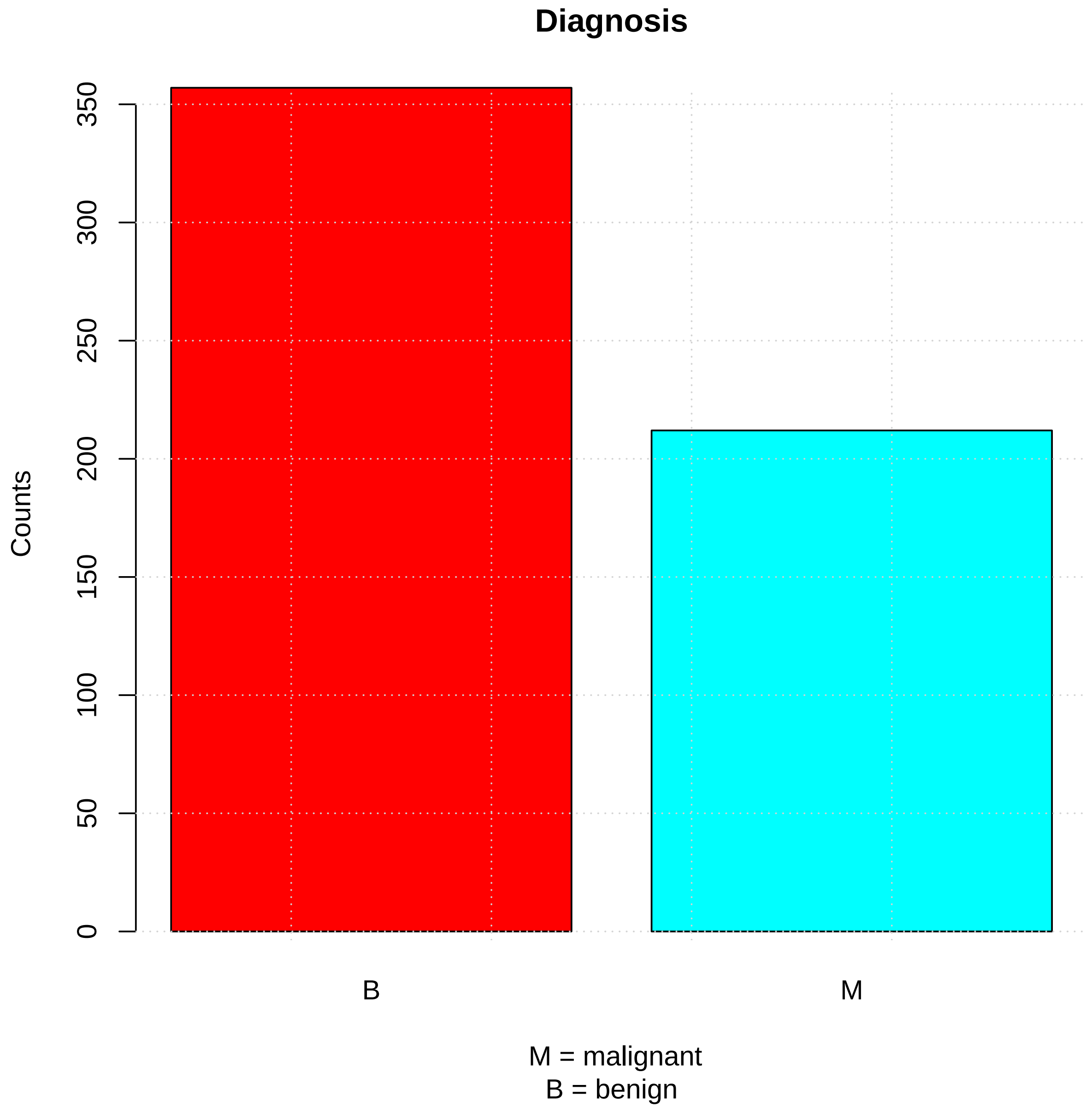

Figure 1: Histogram of qualitative data – Diagnosis.

3.3 Density plots

Below are charts of the distribution of all quantitative features. For all features describing means (figure 2), standard error (figure 3) and "worst"/largest values (figure 4), we do not notice any match with normal distribution. Based only on the graphs, we are unable to identify specific distributions that describe the data to a satisfactory degree.

It is worth mentioning that the values of features describing the average (figure 2) and "worst"/highest values (figure 4) come from different distributions, or from a distribution with different parameters – the density plots only slightly differ in shape. The density plots of the values of features describing the standard error (figure 3) have a similar shape, but definitely vary in the range of values on the OY and OX axis.

Histograms of all variables corresponding to the density plots are available in the file "Medical diagnostics: Breast Cancer Wisconsin (Diagnostic) Project – Additional Plots".

3.3.0.1 Features describing the mean

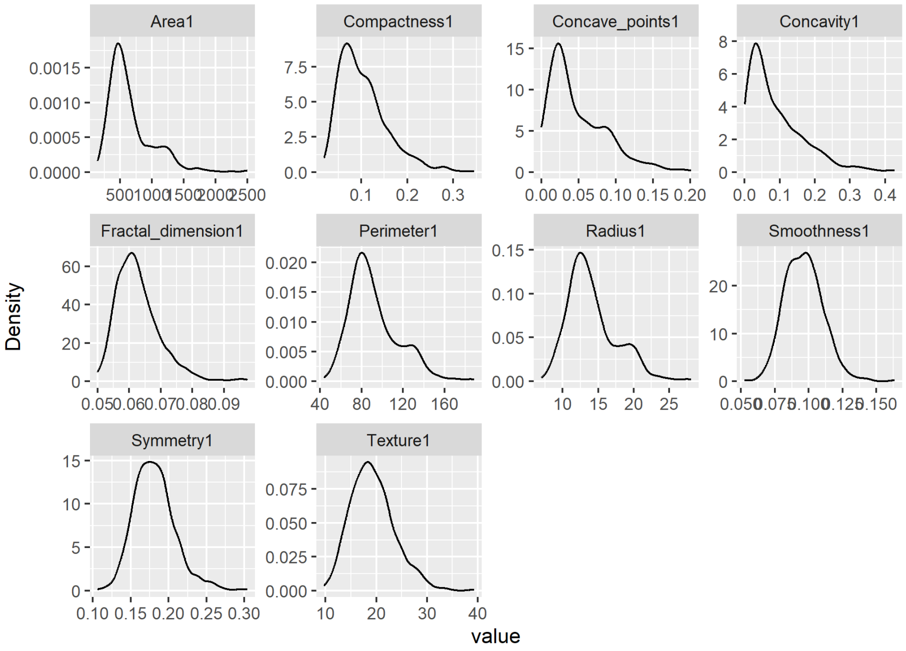

Figure 2: Density plots for features describing the mean.

3.3.0.2 Features describing the standard error

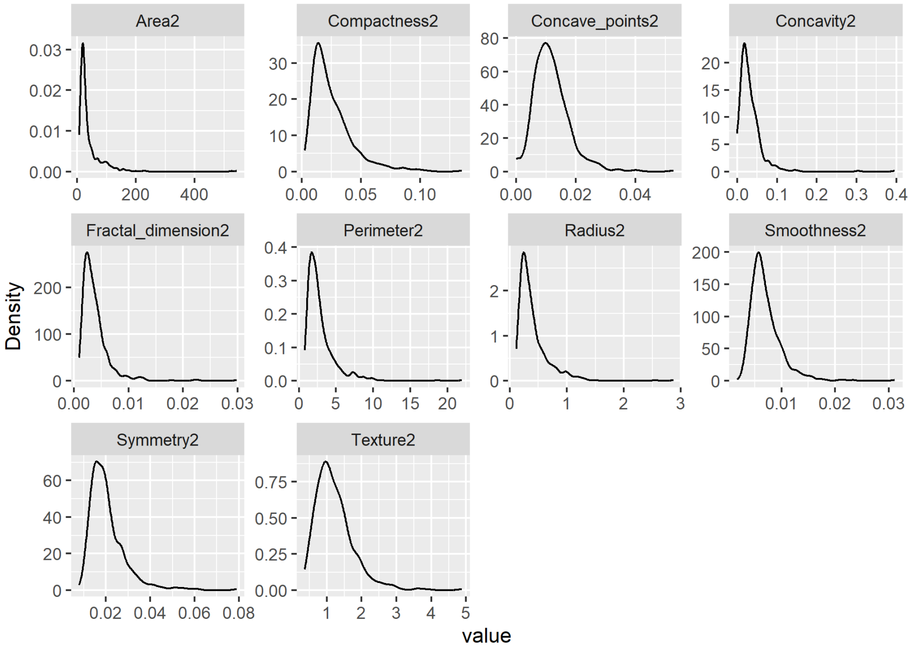

Figure 3: Density plots for features describing the standard error.

3.3.0.3 Features describing the "worst"/largest value

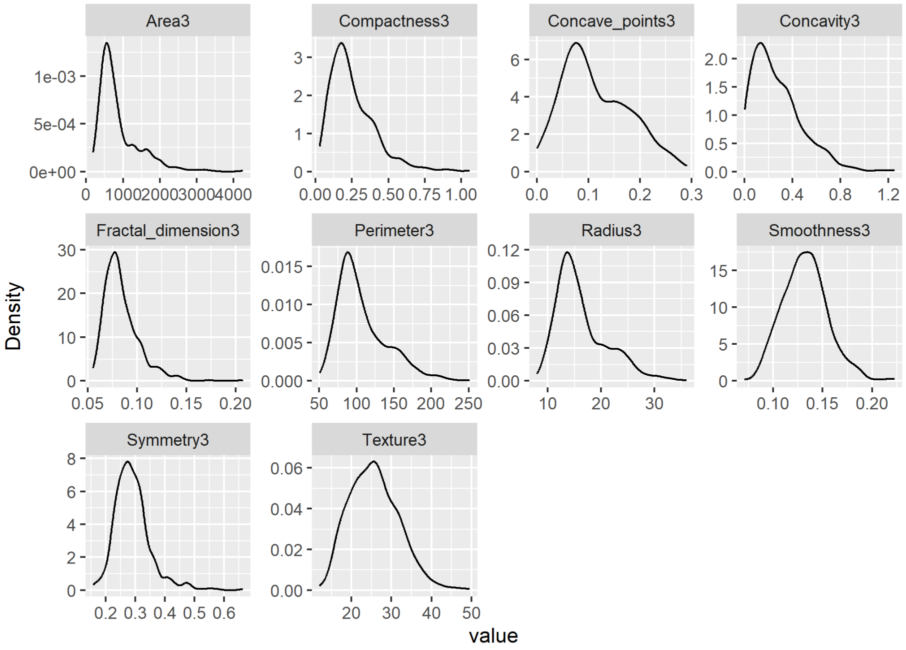

Figure 4: Density plots for features describing the "worst"/largest value.

3.4 Boxplots

Below are boxplots of the values of all quantitative features separated into classes of people with benign and malignant cancer.

3.4.0.1 Features describing the mean

Analyzing the boxplot for the values of features describing the mean (figure 5), we can see that for variables except the Fractal_dimension, each box plot is shifted relative to the class. Individuals with benign cancer have smaller mean values, as well as a smaller interquartile range (for the Texture, Symmetry and Smoothness parameters, these values are similar). In each of the charts, we can also see values that are more than $3 * IQR$ apart, indicating the presence of outliers.

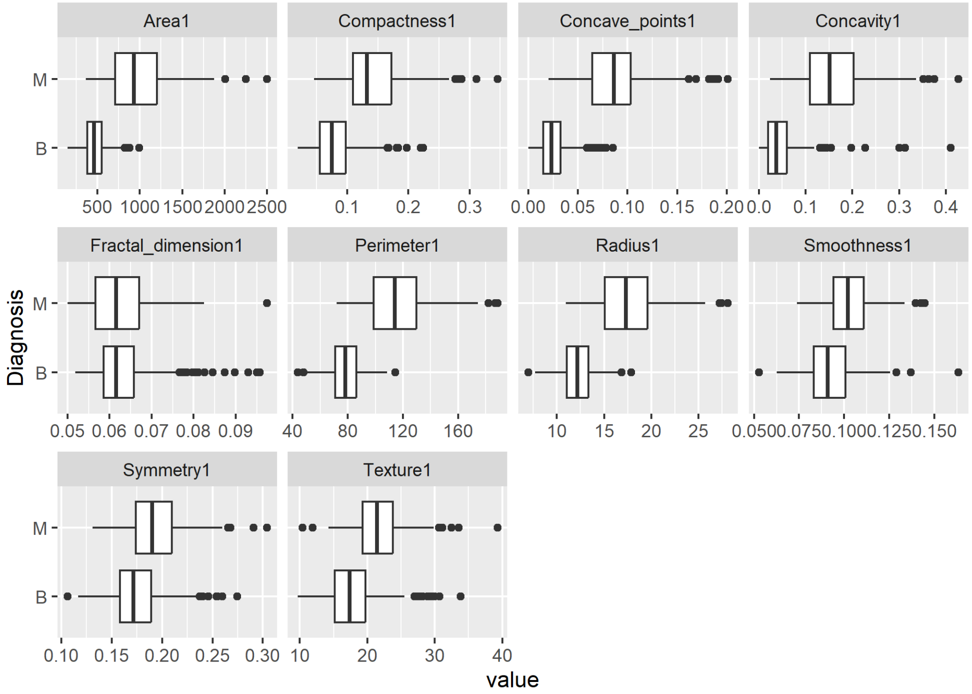

Figure 5: Boxplots for features describing the mean.

3.4.0.2 Features describing the standard error

Analyzing the boxplot for the values of variables describing the standard error (figure 6), we can see that for features other than Smoothness, Symmetry and Texture, each boxplot is shifted relative to the class. Individuals with benign cancer have smaller average values. The interquartile range is also smaller for the features describing Area, Perimeter and Radius; for other features it is rather similar. In each of the graphs, we can also see values that are more than $3 * IQR$ apart, indicating the presence of outliers.

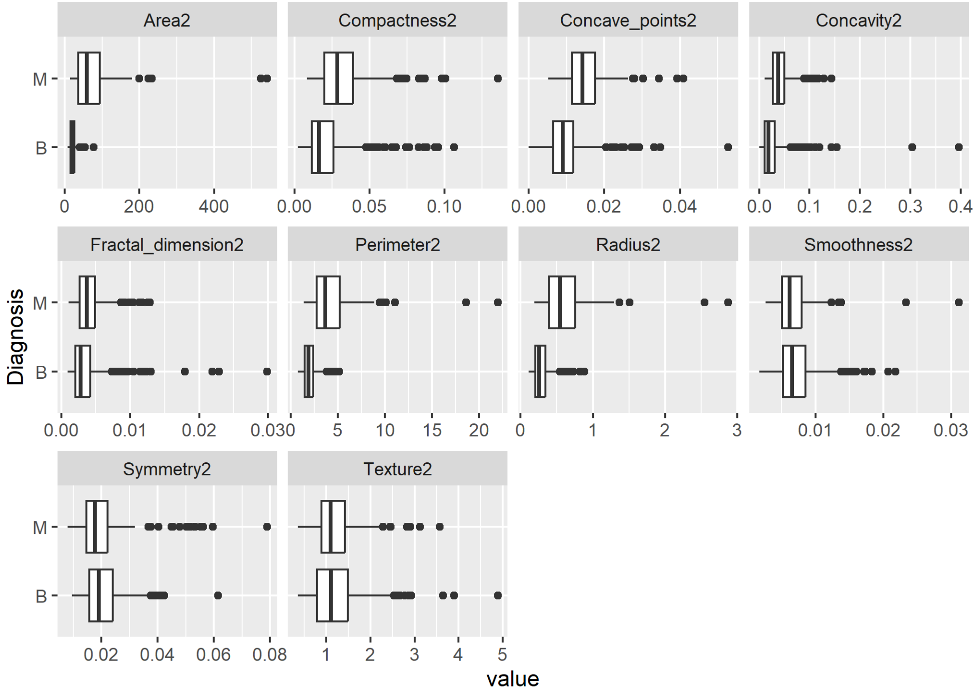

Figure 6: Boxplots for features describing the standard error.

3.4.0.3 Features describing the ”worst”/largest value

Analyzing the boxplot for the features of the variables describing the ”worst”/largest values (figure 7), we can see that for all variables, each boxplot is shifted relative to the class. For these ten features individuals with benign cancer are characterized by smaller mean values, as well as a smaller interquartile range (for variables describing Smoothness and Texture, the interquartile range has a similar value). In each of the graphs, we can also see values that are more than $3 * IQR$ away, indicating the presence of outliers.

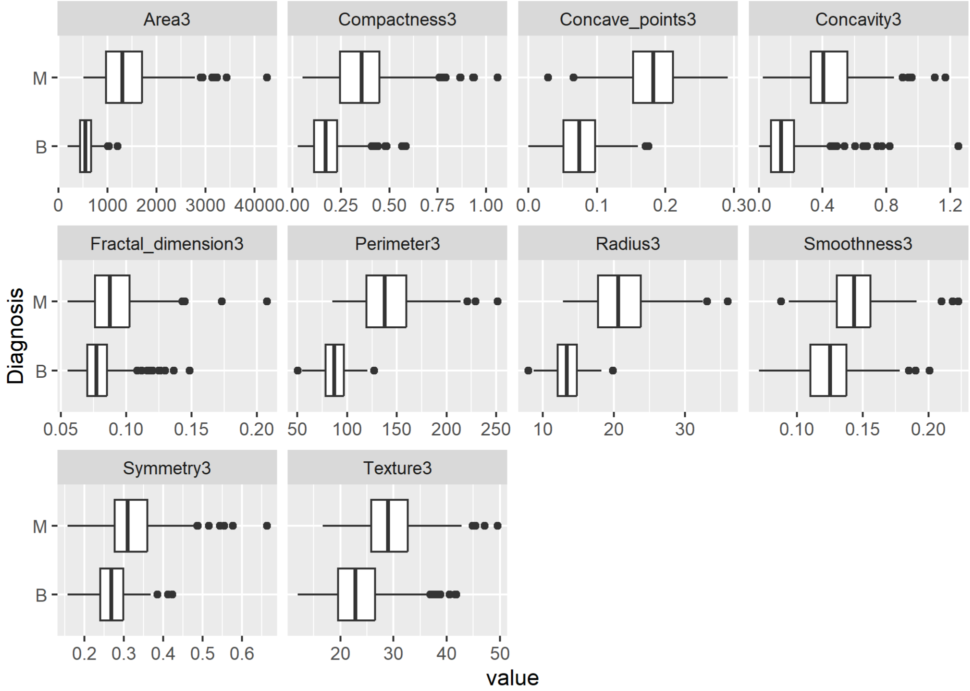

Figure 7: Boxplots for features describing the ”worst”/largest value.

3.5 QQplots

Below are QQplots of the values of all quantitative features. For all features describing means (figure 8), standard error (figure 9) and ”worst”/largest values (figure 10), we do not notice any compatibility with the standard normal distribution. For the variables whose values were nearest to the straight line, that is Smoothness1, Texture1 and Texture3, we conducted the Shapiro-Wilk test. In each case, the null hypothesis assuming that our research sample is drawn from a normal distribution was rejected for p-values at the 0.05 mark.

3.5.0.1 Features describing the mean

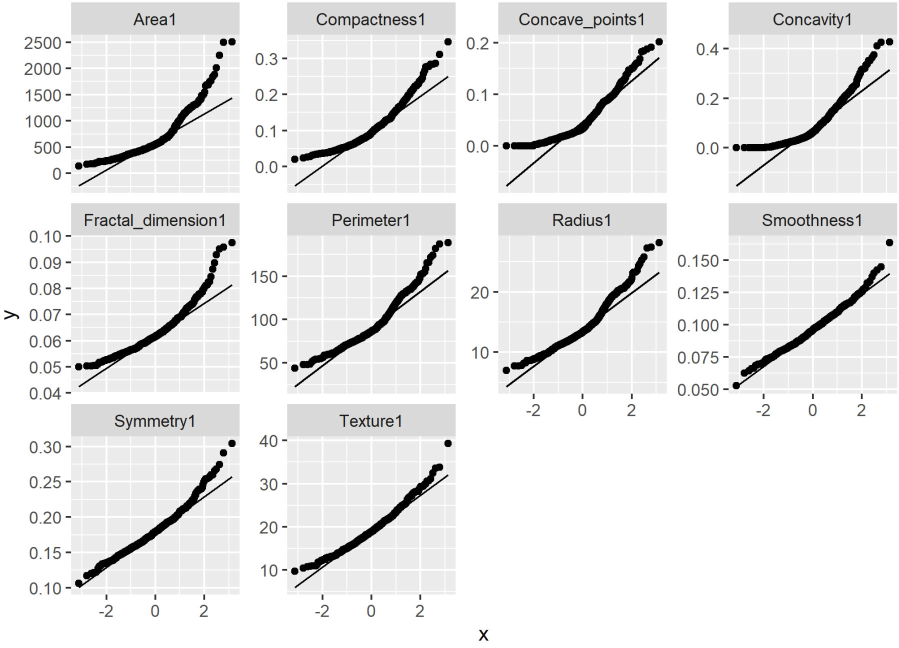

Figure 8: QQplots for features describing the mean.

3.5.0.2 Features describing the standard error

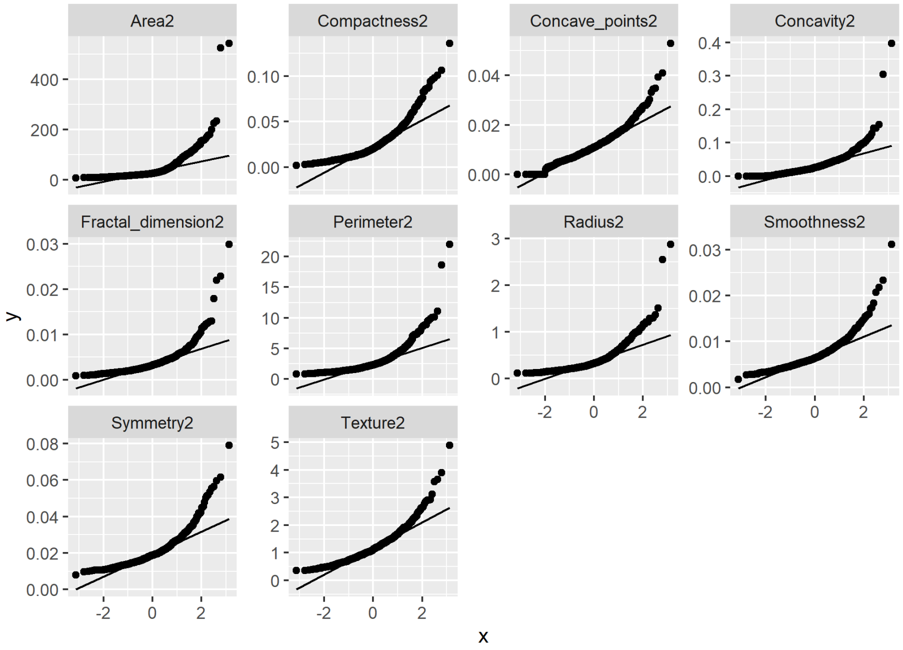

Figure 9: QQplots for features describing the standard error.

3.5.0.3 Features describing the "worst"/largest value

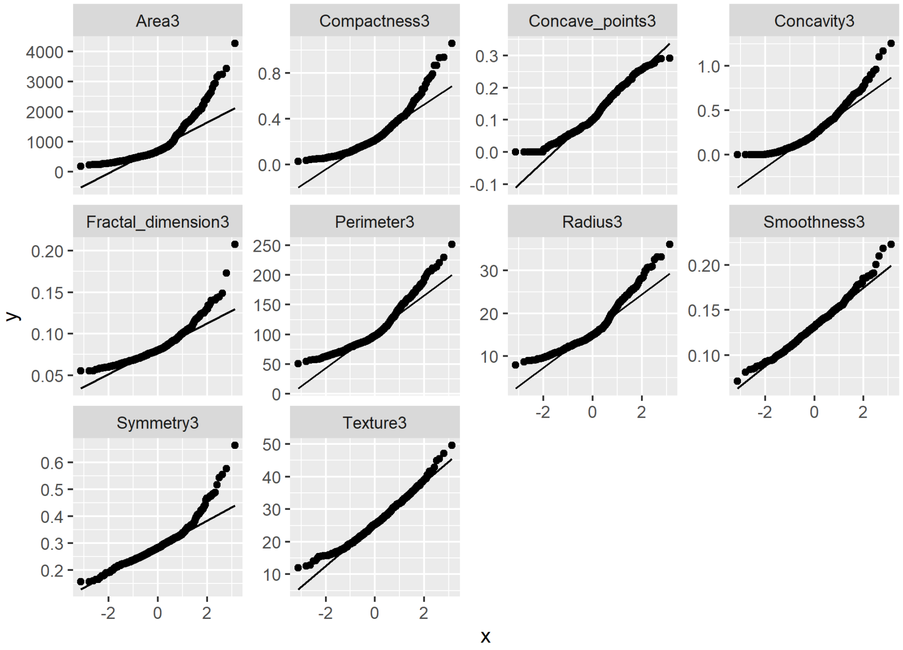

Figure 10: QQplots for features describing the "worst"/largest value.

3.6 Correlation

Below there is a correlation plot between all the quantitative variables available in the dataset (figure 11). In the document "Medical Diagnostics: Breast Cancer Wisconsin (Diagnostic) Project – Additional Plots" one can find correlation charts separated according to the values they describe (that is features describing means, standard errors and "worst"/largest values).

After analyzing the plot, it is straightforward to notice that we are dealing with a number of highly correlated variables. These are (each correlation between two features appears only once):

- **Radius1** with Perimeter1, Area1, Concavity1, Concave_Points1, Radius2, Perimeter2, Area2, Radius3, Perimeter3, Area3, Concave_points3,
- **Texture1** with Texture3,
- **Perimeter1** with Area1, Concavity1, Concave_Points1, Radius2, Perimeter2, Area2, Radius3, Perimeter3, Area3, Concave_points3,
- **Area1** with Concavity1, Concave_Points1, Radius2, Perimeter2, Area2, Radius3, Perimeter3, Area3, Concave_points3,
- **Smoothness1** with Compactness1, Smoothness3,
- **Compactness1** with Concavity1, Concave_Points1, Compactness3, Concavity3, Concave_Points3,
- **Concavity1** with Concave_Points1, Concavity3, Concave_Points3,
- **Concave_Points1** with Radius3, Perimeter3, Concave_Points3,
- **Fractal_dimension1** with Fractal_dimension3,
- **Radius2** with Perimeter2, Area2, Radius3, Perimeter3, Area3,
- **Perimeter2** with Area2, Radius3, Perimeter3, Area3,
- **Area2** with Radius3, Perimeter3, Area3,
- **Compactness2** with Concavity2, Concave_points2, Fractal_dimension2,
- **Radius3** with Perimeter3, Area3, Concave_points3,
- **Perimeter3** with Area3, Concave_points3,
- **Area3** with Concave_points3,
- **Compactness3** with Concavity3, Concave_points3, Fractal_dimension3,
- **Concavity3** with Concave_points3.

Therefore, further into the project, we will limit the number of features that we will use to create models for classification. To reduce the number of features, we will use functions available in R.

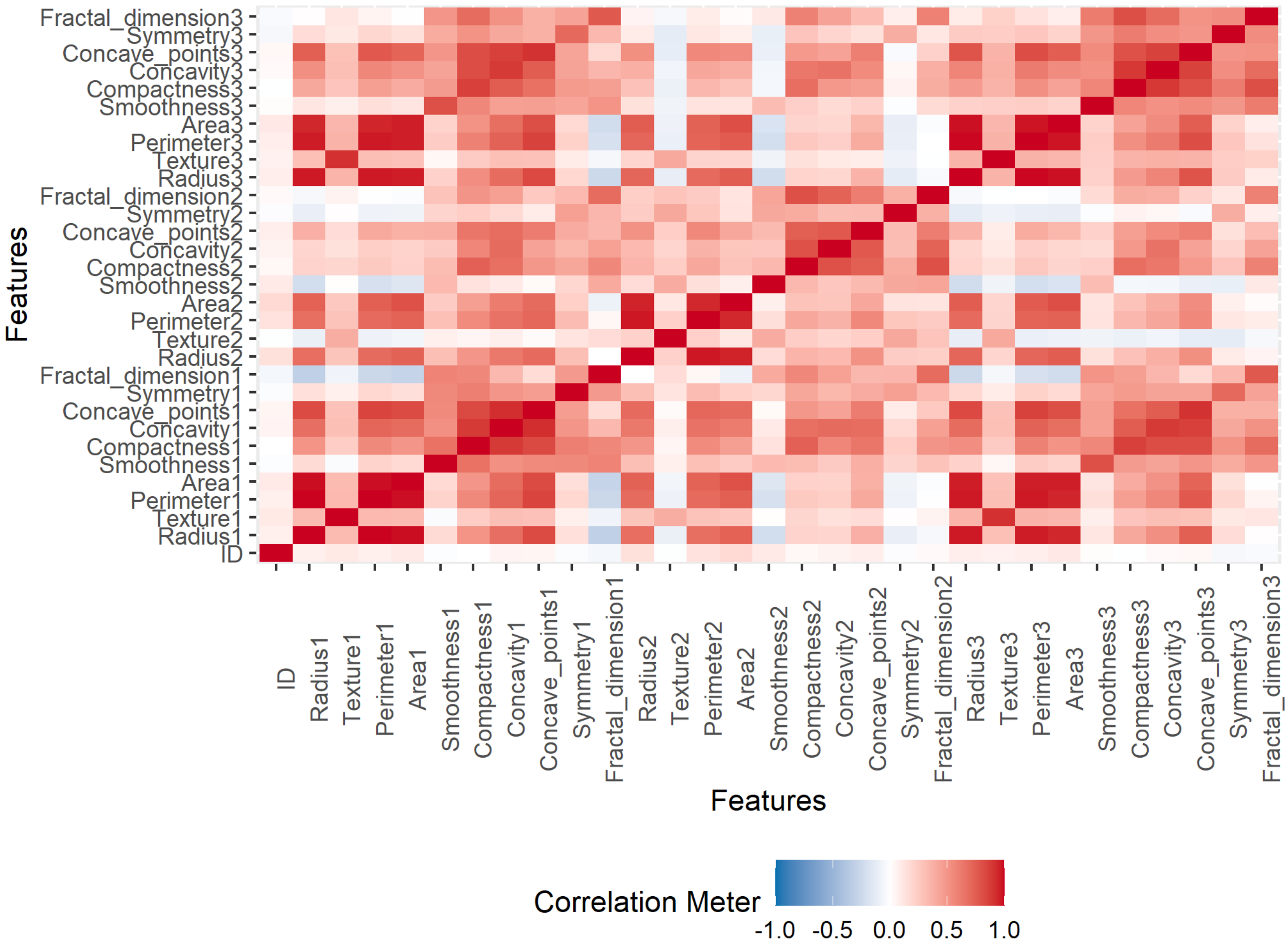

Figure 11: Correlation of all features.

3.7 Scatter plots

Below are two sample scatter plots for the variables describing the mean values and standard errors of Radius, Texture , Perimeter, Area, Smoothness (figure 12) and the variables Radius1, 2 and 3 (figure 13) broken down by class – people with benign and malignant cancer. In the document ”Medical Diagnostics: Breast Cancer Wisconsin (Diagnostic) Project – Additional Plots” one can find another scatter plots, that we analyzed.

For each of the subsequent plots, it shows how any two features are able to separate the two considered classes. However, in all of the charts, we don’t see a clear dominance of the two features that perfectly classify the patients – in each plot, the classes partially intersect. What also works to our disadvantage is that all of the partial graphs look very similar. This prevents us from selecting features for the model in advance. In addition, in many of the graphs we can see the strong correlation described earlier.

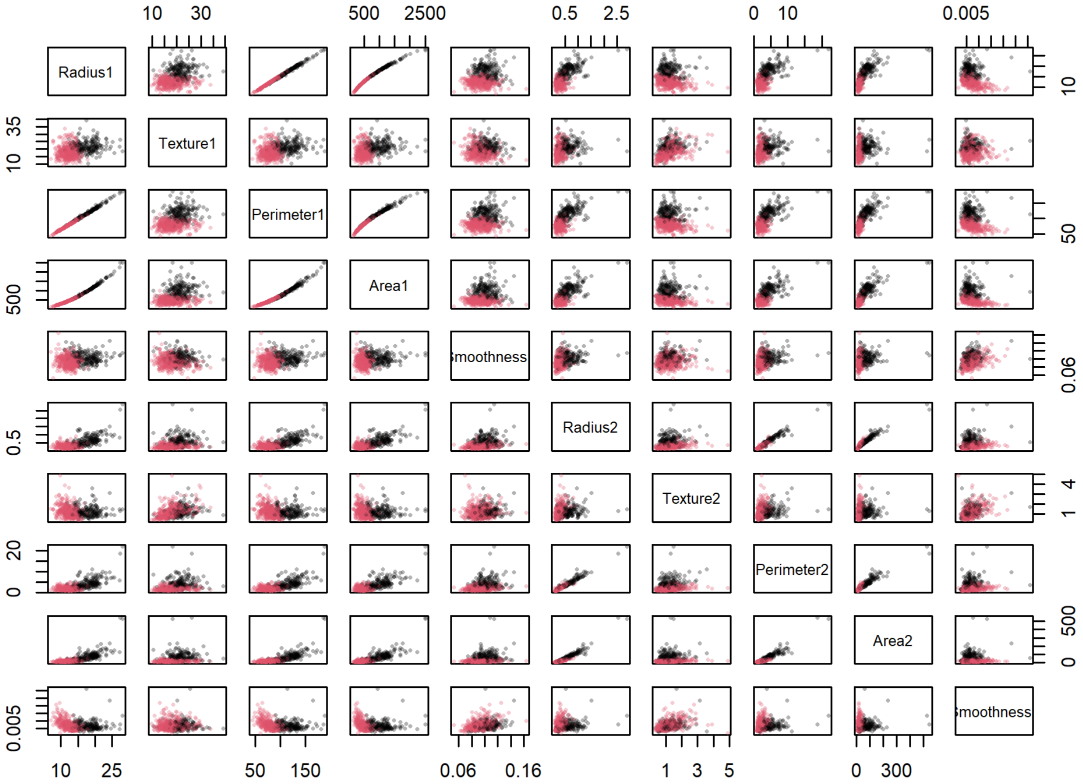

Figure 12: A matrix of scatter plots for Radius1, Texture1, Perimeter1, Area1, Smoothness1, Radius2, Texture2, Perimeter2, Area2, Smoothness2.

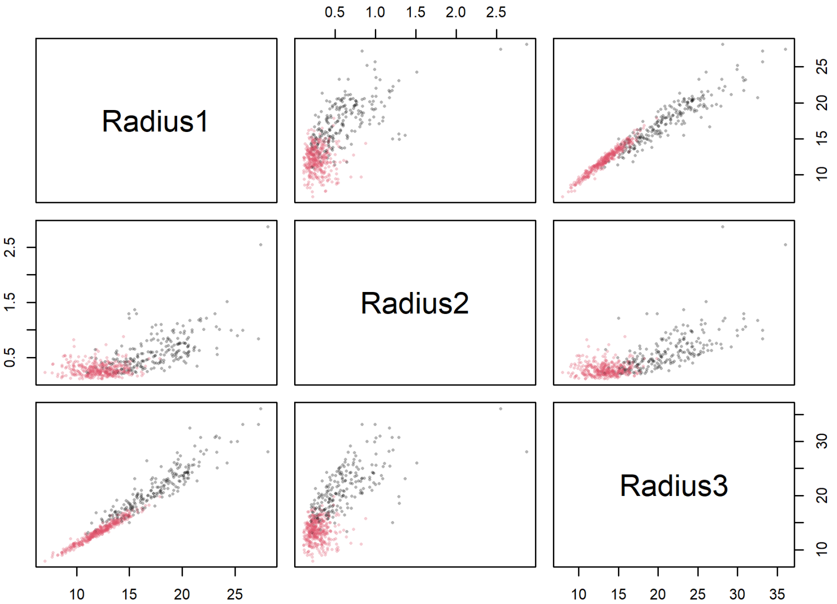

Figure 13: A matrix of scatter plots for Radius1, Radius2, Radius3.

4 Classification

4.1 Check correlations

First, we identify highly correlated variables and remove columns to reduce pairwise correlations. In further analysis, we will consider data without columns:

```
##  [1] "Concavity1"      "Concave_points1" "Perimeter3"      "Radius3"
##  [5] "Perimeter1"      "Area3"           "Radius1"         "Perimeter2"
##  [9] "Area2"           "Texture1"
```

4.2 Learning and test set split

To split the data into two sets, we randomly select 2/3 of the data into the learning set and assign the remaining observations to the testing set. Then we check the proportions of classes in the training set (first line) and in the test set (second line).

```
##
##         B         M
## 0.6253298 0.3746702
##
##         B         M
## 0.6315789 0.3684211
```

Comparing the above proportions with the class proportions for the data,

```
##
##         B         M
## 0.6274165 0.3725835
```

we can see that the class proportions after the split have been preserved.

4.3 Features selections

Using the stepwise method for the training set, we select various combinations of features for which we will create models. We use the *stepclass* function with the *forward* direction for three methods: *lda*, *qda* and *sknn*. We get the following results

1. $\boldsymbol{Concave\_points3 + Area1 + Texture3 + Radius2 + Fractal\_dimension2}$ for the linear discriminant analysis method,

2. $\boldsymbol{Concave\_points3 + Area1 + Texture3 + Concave\_points2 + Smoothness3}$ for the quadratic discriminant analysis method,

3. $\boldsymbol{Concave\_points3 + Radius2 + Concavity3 + Compactness1}$ for the k-nearest neighbors method.

Additionally, as the fourth subset, we will consider the set of **all features** obtained after removing correlations.

4.4 Classification rules

For all four features subsets mentioned above, we construct classification rules based on the selected variables. We build 7 different classification models for the learning test, using:

- linear discriminant analysis (lda),
- quadratic discriminant analysis (qda),
- 5-nearest neighbors method (knn),
- logistic regression (logit),
- classification tree (tree),
- bagging method (bagging),
- random forest (randomForest).

Next, we predict class labels for test set on the basis of learning set.

4.5 Assessment of classification accuracy

4.5.1 Confusion matrices

We then create confusion matrices for all models used. The columns contain the real values of the labels, while the rows contain their predictions.

1. $Concave\_points3 + Area1 + Texture3 + Radius2 + Fractal\_dimension2$

```
##                 real.labels
## pred.labels.lda1   B   M
##                B 120   8
##                M   0  62
##                 real.labels
## pred.labels.qda1   B   M
##                B 119   7
##                M   1  63
##                 real.labels
## pred.labels.knn1   B   M
##                B 113  15
##                M   7  55
##                   real.labels
## pred.labels.logit1   B   M
##                  B 119   5
##                  M   1  65
##                  real.labels
## pred.labels.tree1   B   M
##                 B 107   1
##                 M  13  69
##                     real.labels
## pred.labels.bagging1   B   M
##                    B 116   3
##                    M   4  67
##                          real.labels
## pred.labels.randomForest1   B   M
##                         B 117   5
##                         M   3  65
```

2. $Concave\_points3 + Area1 + Texture3 + Concave\_points2 + Smoothness3$

```
##                 real.labels
## pred.labels.lda2   B   M
##                B 120   7
##                M   0  63
##                 real.labels
## pred.labels.qda2   B   M
##                B 118   5
##                M   2  65
##                 real.labels
## pred.labels.knn2   B   M
##                B 113  15
##                M   7  55
##                   real.labels
## pred.labels.logit2   B   M
##                  B 117   7
##                  M   3  63
##                  real.labels
## pred.labels.tree2   B   M
##                 B 113   5
##                 M   7  65
##                     real.labels
## pred.labels.bagging2   B   M
##                    B 114   2
##                    M   6  68
##                          real.labels
## pred.labels.randomForest2   B   M
##                         B 115   3
##                         M   5  67
```

3. $Concave\_points3 + Radius2 + Concavity3 + Compactness1$

```
##                 real.labels
## pred.labels.lda3   B   M
##                B 119   9
##                M   1  61
##                 real.labels
## pred.labels.qda3   B   M
##                B 114  11
##                M   6  59
##                 real.labels
## pred.labels.knn3   B   M
##                B 111  11
##                M   9  59
##                   real.labels
## pred.labels.logit3   B   M
##                  B 119   9
##                  M   1  61
##                  real.labels
## pred.labels.tree3   B   M
##                 B 106   7
##                 M  14  63
##                     real.labels
## pred.labels.bagging3   B   M
##                    B 114  11
##                    M   6  59
##                          real.labels
## pred.labels.randomForest3   B   M
##                         B 115   8
##                         M   5  62
```

4. All features

```
##                 real.labels
## pred.labels.lda4   B   M
##                B 119   8
##                M   1  62
##                 real.labels
## pred.labels.qda4   B   M
##                B 111   6
##                M   9  64
##                  real.labels
## pred.labels.knn4   B   M
##                B 113  15
##                M   7  55
##                   real.labels
## pred.labels.logit4   B   M
##                  B 117   4
##                  M   3  66
##                  real.labels
## pred.labels.tree4   B   M
##                 B 107   1
##                 M  13  69
##                     real.labels
## pred.labels.bagging4   B   M
##                    B 115   4
##                    M   5  66
##                         real.labels
## pred.labels.randomForest4   B   M
##                         B 118   3
##                         M   2  67
```

It is noticeable that none of the classifiers was able to predict label values for every model under consideration with 100% accuracy. Nevertheless, each matrix's diagonal has significantly greater values, suggesting that the classes under consideration have been classified rather successfully. Below, we'll examine each classifier's accuracy.

4.5.2 Accuracy

Based on the above matrices, we calculate and compare the accuracy. The results are presented below and all values are rounded to three decimal places.

```
##                        1 features subset 2 features subset 3 features subset
## accuracy.lda                      95.789            96.316            94.737
## accuracy.qda                      95.789            96.316            91.053
## accuracy.knn                      88.421            88.421            89.474
## accuracy.logit                    96.842            94.737            94.737
## accuracy.tree                     92.632            93.684            88.947
## accuracy.bagging                  96.316            95.789            91.053
## accuracy.randomForest             95.789            95.789            93.158
##                        4 features subset
## accuracy.lda                      95.263
## accuracy.qda                      92.105
## accuracy.knn                      88.421
## accuracy.logit                    96.316
## accuracy.tree                     92.632
## accuracy.bagging                  95.263
## accuracy.randomForest             97.368
```

The studied classes were effectively separated by each classifier. It is challenging to determine which one performed best for all the features sets under examination - almost all of them have accuracy values above 91%. The worst outcomes, however, were obtained by the tree classifier for the third set of characteristics and the knn classifier for all four features subsets. Their accuracy is only $88 - 89\%$. We achieve the best results for the random forest classifier and all features. Its accuracy is more than 97%.

In conclusion, the following classifiers are the most successful for each features subset:

- first features subset – logistic regression classifier with 96.842% accuracy,
- second features subset – lda and qda classifiers with 96.316% accuracy,
- third features subset – lda and logistic regression classifiers with 94.737% accuracy,
- all features – the random forest classifier with 97.368% accuracy.

4.6 Cross-validation

Then we repeat the above steps using a 10-fold cross-validation scheme. We repeatedly draw training and test sets and build classifiers on the training set, check them on the test set and average the results.

4.6.1 Accuracy

The accuracy results are presented below and all values are rounded to three decimal places.

```
##                          1 features subset 2 features subset 3 features subset
## accuracy.lda.cv                     95.782            95.079            93.146
## accuracy.qda.cv                     95.606            95.782            91.564
## accuracy.knn.cv                     89.982            89.104            92.619
## accuracy.logit.cv                   96.661            96.485            93.497
## accuracy.tree.cv                    94.728            94.376            92.443
## accuracy.bagging.cv                 94.025            95.782            93.497
## accuracy.randomForest.cv            95.606            96.309            92.619
##                          4 features subset
## accuracy.lda.cv                     96.134
## accuracy.qda.cv                     94.903
## accuracy.knn.cv                     89.104
## accuracy.logit.cv                   96.309
## accuracy.tree.cv                    94.200
## accuracy.bagging.cv                 95.782
## accuracy.randomForest.cv            95.606
```

Again, each classifier efficiently identified the studied classes. The accuracy of almost all of them for each of the considered features sets is greater than 91%. The knn classifier performed the worst also in this case. For the first, second and fourth set of features, its accuracy is only 89%. However, for each subset of features, the most effective classifier turned out to be the logistic regression classifier.

5 Classification on standardized data

5.1 Data sets, features selection and classification rules

Again, we consider the same learning and test set as in section 4, without performing cross-validation. This time we standardize all of our quantitative features. We will again perform the analysis for the same subsets described in section 4.3. For all four features subsets, we construct classification rules based on the selected variables. We build 7 different classification models for the learning test, using all classifiers described in section 4.4.

5.2 Assessment of classification accuracy

5.2.1 Confusion matrices

First, we create confusion matrices. The columns contain the real values of the labels, while the rows contain their predictions.

1. $Concave\_points3 + Area1 + Texture3 + Radius2 + Fractal\_dimension2$

```
##                   real.labels
## pred.labels.lda1.s   B   M
##                  B 120   8
##                  M   0  62
##                   real.labels
## pred.labels.qda1.s   B   M
##                  B 119   7
##                  M   1  63
##                   real.labels
## pred.labels.knn1.s   B   M
##                  B 118  11
##                  M   2  59
##                     real.labels
## pred.labels.logit1.s   B   M
##                    B 119   5
##                    M   1  65
##                    real.labels
## pred.labels.tree1.s   B   M
##                   B 107   1
##                   M  13  69
##                       real.labels
## pred.labels.bagging1.s   B   M
##                      B 117   2
##                      M   3  68
##                            real.labels
## pred.labels.randomForest1.s   B   M
##                           B 118   5
##                           M   2  65
```

2. $Concave\_points3 + Area1 + Texture3 + Concave\_points2 + Smoothness3$

```
##                   real.labels
## pred.labels.lda2.s   B   M
##                  B 120   7
##                  M   0  63
##                   real.labels
## pred.labels.qda2.s   B   M
##                  B 118   5
##                  M   2  65
##                   real.labels
## pred.labels.knn2.s   B   M
##                  B 119   7
##                  M   1  63
##                     real.labels
## pred.labels.logit2.s   B   M
##                    B 117   7
##                    M   3  63
##                    real.labels
## pred.labels.tree2.s   B   M
##                   B 113   5
##                   M   7  65
##                       real.labels
## pred.labels.bagging2.s   B   M
##                      B 113   3
##                      M   7  67
##                            real.labels
## pred.labels.randomForest2.s   B   M
##                           B 115   2
##                           M   5  68
```

3. $Concave\_points3 + Radius2 + Concavity3 + Compactness1$

```
##                   real.labels
## pred.labels.lda3.s   B   M
##                  B 119   9
##                  M   1  61
##                   real.labels
## pred.labels.qda3.s   B   M
##                  B 114  11
##                  M   6  59
##                   real.labels
## pred.labels.knn3.s   B   M
##                  B 113   5
##                  M   7  65
##                     real.labels
## pred.labels.logit3.s   B   M
##                    B 119   9
##                    M   1  61
##                    real.labels
## pred.labels.tree3.s   B   M
##                   B 106   7
##                   M  14  63
##                        real.labels
## pred.labels.bagging3.s   B   M
##                       B 114  11
##                       M   6  59
##                             real.labels
## pred.labels.randomForest3.s   B   M
##                           B 115   9
##                           M   5  61
```

4. All features

```
##                   real.labels
## pred.labels.lda4.s   B   M
##                  B 119   8
##                  M   1  62
##                   real.labels
## pred.labels.qda4.s   B   M
##                  B 111   6
##                  M   9  64
##                   real.labels
## pred.labels.knn4.s   B   M
##                  B 119   9
##                  M   1  61
##                     real.labels
## pred.labels.logit4.s   B   M
##                    B 117   4
##                    M   3  66
##                    real.labels
## pred.labels.tree4.s   B   M
##                   B 107   1
##                   M  13  69
##                        real.labels
## pred.labels.bagging4.s   B   M
##                       B 112   4
##                       M   8  66
##                             real.labels
## pred.labels.randomForest4.s   B   M
##                           B 118   3
##                           M   2  67
```

It can be easily seen that no classifier was perfectly predictive of label values for each of the models considered. That said, there are significantly larger values on the diagonal of each matrix, indicating a relatively successful classification of the classes being considered. We will analyze accuracy of all classifiers below.

5.2.2 Accuracy

Based on the above matrices, we calculate and compare the accuracy. The results are presented below and all values are rounded to three decimal places.

```
##                         1 features subset 2 features subset 3 features subset
## accuracy.lda.s                     95.789            96.316            94.737
## accuracy.qda.s                     95.789            96.316            91.053
## accuracy.knn.s                     93.158            95.789            93.684
## accuracy.logit.s                   96.842            94.737            94.737
## accuracy.tree.s                    92.632            93.684            88.947
## accuracy.bagging.s                 97.368            94.737            91.053
## accuracy.randomForest.s            96.316            96.316            92.632
##                         4 features subset
## accuracy.lda.s                     95.263
## accuracy.qda.s                     92.105
## accuracy.knn.s                     94.737
## accuracy.logit.s                   96.316
## accuracy.tree.s                    92.632
## accuracy.bagging.s                 93.684
## accuracy.randomForest.s            97.368
```

Each of the classifiers separated the studied classes well. It is rather difficult to determine one that performed best for each of the considered features sets – the accuracy of almost all of them is greater than 91%. However, the tree classifier for the third model performed the worst. Its accuracy is only 89%. We achieve the best results for the bagging classifier and model 1, as well as for the random forest and the model containing all features. Their accuracy is as high as 97%.

In summary, the best classifiers for every features subsets are:

- first features subset – bagging classifier with 97.368% accuracy,
- second features subset – lda, qda and random forest classifiers with 96.316% accuracy,
- third features subset – lda and logit classifiers with 94.737% accuracy,
- all features – the random forest classifier with 97.368% accuracy.

5.2.3 Accuracy difference between standardized and non-standardized data

Below is a table of accuracy difference between classifiers based on standardized and non-standardized data. All differences are calculated by subtracting the accuracy for standardized data from those for non-standardized data, rounded to three decimal places.

```
##                       1 features subset 2 features subset 3 features subset
## accuracy.lda                      0.000             0.000             0.000
## accuracy.qda                      0.000             0.000             0.000
## accuracy.knn                     -4.737            -7.368            -4.210
## accuracy.logit                    0.000             0.000             0.000
## accuracy.tree                     0.000             0.000             0.000
## accuracy.bagging                 -1.052             1.052             0.000
## accuracy.randomForest            -0.527            -0.527             0.526
##                       4 features subset
## accuracy.lda                        0.000
## accuracy.qda                        0.000
## accuracy.knn                       -6.316
## accuracy.logit                      0.000
## accuracy.tree                       0.000
## accuracy.bagging                    1.579
## accuracy.randomForest               0.000
```

As can be seen, data standardization has increased accuracy of models in several cases. These are:

- **5-nn classifier** — for all analyzed models,
- **bagging classifier** — for first model,
- **random forest** — for first and second model.

Unfortunately, the accuracy decreased in three cases: for bagging classifier in second and fourth model and for random forest in third model. For all other cases, nothing has changed.

That said, we can see that standardizing the data for the knn (in this case for 5 neighbours) improved the results. For non-standardized data, for this method, the accuracy for each subset of features did not even exceed 90%, but for standardized data it is $\sim 93 - 95\%$.

6 Discussion

In this paper, we have shown that each of the 7 proposed classifiers effectively separate the studied classes for 4 different sets of features. Almost all of them have an accuracy above 91% for all feature subsets and it is rather difficult to clearly determine which one of the classifiers is the best or which set of parameters give the most successful outcomes.

The best classifiers for each of the feature subsets considered have accuracies around $95 - 97\%$. Classifiers with the highest accuracy scores are: the classifier based on logistic regression, lda, qda and random forest. Nevertheless, the worst results are achieved by the knn classifier (for 5 neighbours). Its accuracy is usually around only $88 - 89\%$. However, this classifier's performance can be improved by standardizing the data.

The following might be taken into consideration as an extension of the work done in this project:

- changing the parameters of classification methods,
- examining other feature subsets,
- the selection of additional classification techniques.
- identyfing and removing outliers before performing classification.

7 Bibliography

1. my.clevelandclinic.org
2. Data Mining Course materials

3. "Medical diagnostics: Breast Cancer Wisconsin (Diagnostic) Project – Additional Plots" – P.Cieślachowska, A. Winiarska