Data Mining Project, Part 1
Breast Cancer

Tymoteusz Cieślik, Jakub Błażejewski

30.11.2021

1 Introduction and description of the data

With no surprise, we all can say that people born and die everyday. Over the years, the medical system developed significantly, and so the births and deaths look different from hundreds of years ago. With world's progress, we can see that there are more and more dangerous diseases that are evolving and unfortunately there is still no cure for all of them. As the Central Statistical Office (pl. *Główny Urząd Statystyczny*) claims, cancer is placed second of the reasons of deaths in Poland. When early detected, cancer can be treated and people get better, but not everyone has that much luck.

Breast cancer is one of the most dangerous diseases for women (for men also, but mainly women). It begins when healthy cells in the breast change and grow out of control, forming a mass or sheet of cells called a tumor. A tumor can be cancerous or benign. A cancerous tumor is malignant, meaning it can grow and spread to other parts of the body. A benign tumor means the tumor can grow but will not spread. As it is very important topic, we decided to deepen our knowledge about breast cancer, its features and behaviour by analysing the dataset provided by UCI Machine Learning called *Breast Cancer Wisconsin (Diagnostic)*. The features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass.

Before starting the proper analysis, it is always good to know with what data we work with. Below you can find the basic information about the data set.

Data characteristics:

- data size:
  - N (all) = 569
  - B = 357
  - M = 212
- features:
  - 1 categorical
  - 10 numerical
- missing values: None

Analysed data set contains following information about the tumor, and each of the numerical variables has 3 different types in the dataset – its mean, standard error, and worst parameter:

- ID number,
- diagnosis (categorical)
  - B - benign,
  - M - malignant.

- features (numerical):
  - radius - mean of distances from center to points on the perimeter,
  - texture - standard deviation of gray-scale values,
  - perimeter,
  - area,
  - smoothness - local variation in radius lengths,
  - compactness - ($\frac{\text{perimeter}^2}{area}$ - 1),
  - concavity - severity of concave portions of the contour,
  - concave points - number of concave portions of the contour,
  - symmetry,
  - fractal dimension - (coastline approximation - 1).

Additionally, in our analysis we use term of "the worst", which means three largest value in each feature.

2 Theory and used methods

Most of the methods we used for our analysis are those learnt on the course Data Mining. Specifically they were:

- data preprocessing
  - finding missing values
  - finding outliers
  - grouping the data
  - assessing which factors affect the diagnosis to the greater extent
- classification methods
  - logistic regression
  - k nearest neighbours
  - Linear Discriminant Analysis
  - Quadratic Discriminant Analysis
  - classification trees
  - bagging
  - support vector machines
- classification accuracy assesment
  - misclassification error
  - confusion matrix
  - ROC Curve
  - AUC (Area Under ROC Curve)

2.1 Logistic Regression

This method uses a logistic function to model a binary dependent variable and is used to model the probability of a certain class or event. Logistic regression is a useful analysis method for classification problems, where you the problem consists of determining if a new sample fits best into a category. Its advantage over the popular linear regression is that it does not require a linear relationship between inputs and output variables. This is due to applying a nonlinear log transformation to the odds ratio:

$$LR = \frac{1}{1 + e^{-x}}.$$

The logistic regression model itself simply models probability of output in terms of input and does not perform statistical classification, though it can be used to make a classifier, for instance by choosing a cutoff value and classifying inputs with probability greater than the cutoff as one class, below the cutoff as the other. Mathematically, logistic regression estimates a multiple linear regression function defined as:

$$logit(p) = log\left(\frac{p(y=1)}{1-p(y=1)}\right) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{in},$$

for $i = 1 \ldots n$.

2.2 k Nearest Neighbours

This algorithm is a non-parametric classification method used for classification and regression. Its input consists of the k closest training examples in a data set. The output might be depending on whether its classification or regression and takes the form of class membership or the property value for the object, respectively. K-NN is a type of classification where the function is only approximated locally, and all computation is deferred until function evaluation. In the classification phase, k is a user-defined constant, and a test point is classified by assigning the label which is most frequent among the k training samples nearest to that point. The best choice of k depends upon the data, generally, larger values of k reduce effect of the noise on the classification, but make boundaries between classes less distinct. The algorithm is especially useful when the relationship between the explanatory and explained variables is complex or unusual, i.e. difficult to model in the classical way. If the relationship is easy to interpret and the set does not contain outliers, classical methods will usually give more accurate results.

2.3 Linear Discriminant Analysis

This method is used to find a linear combination of features that characterizes or separates two or more classes of objects. Discriminant analysis is used when groups are known a priori. Each case must have a score on one or more quantitative predictor measures, and a score on a group measure. LDA approaches the problem of classification by assuming that the conditional probability density functions $p(\vec{x}|y=0)p(\vec{x}|y=0)$ and $p(\vec{x}|y=1)p(\vec{x}|y=1)$ are both the normal distribution with mean and covariance parameters $(\vec{\mu}_0, \Sigma_0)\,(\vec{\mu}_0, \Sigma_0)$ and $(\vec{\mu}_1, \Sigma_1)\,(\vec{\mu}_1, \Sigma_1)$, respectively. Under this assumption, the Bayes optimal solution is to predict points as being from the second class if the log of the likelihood ratios is bigger than some threshold T, so that:

$$(\vec{x} - \vec{\mu}_0)^T \Sigma_0^{-1} (\vec{x} - \vec{\mu}_0) + \ln|\Sigma_0| - (\vec{x} - \vec{\mu}_1)^T \Sigma_1^{-1} (\vec{x} - \vec{\mu}_1) - \ln|\Sigma_1| \; > \; T.$$

Without any further assumptions, the resulting classifier is referred to as Quadratic Discriminant Analysis. LDA instead makes the additional simplifying homoscedasticity assumption and that the covariances have full rank. In this case, several terms cancel and the above decision criterion becomes a threshold on the dot product: $\vec{w} \cdot \vec{x} > c$. This means that the criterion of an input $\vec{x}$ being in a class y is purely a function of this linear combination of the known observations.

2.4 Quadratic Discriminant Analysis

This method is closely related to linear discriminant analysis, where it is assumed that the measurements from each class are normally distributed. Unlike LDA however, in QDA there is no assumption that the covariance of each of the classes is identical. When the normality assumption is true, the best possible test for the hypothesis that a given measurement is from a given class is the likelihood ratio test. Suppose there are only two groups, with means μ_0, μ_1 and covariance matrices Σ_0, Σ_1 corresponding to $y = 0$ and $y = 1$ respectively. Then the likelihood ratio is given by

$$\text{Likelihood ratio} = \frac{\sqrt{2\pi|\Sigma_1|}^{-1} \exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)\right)}{\sqrt{2\pi|\Sigma_0|}^{-1} \exp\left(-\frac{1}{2}(x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0)\right)} < t$$

for some threshold t. After some rearrangement, it can be shown that the resulting separating surface between the classes is a quadratic. The sample estimates of the mean vector and variance-covariance matrices will substitute the population quantities in this formula.

2.5 Classification Tree

A classification tree is a structural mapping of binary decisions that lead to a decision about the class of an object. Although sometimes referred to as a decision tree, it is more properly a type of decision tree that leads to categorical decisions. Generally, such tree is composed of branches that represent attributes, while the leaves represent decisions. In use, the decision process starts at the trunk and follows the branches until a leaf is reached. As the algorithm lasts, every possible split of the tree is tried and considered, and the best split is the one that produces the largest decrease in diversity of the classification label within each partition. This is repeated for all fields, and the winner is chosen as the best splitter for that node. The process is continued at subsequent nodes until a full tree is generated. Most of the time such tree has many branches and to simplify and optimize the result, it's possible to prune the tree. Pruning is the process of removing leaves and branches to improve the performance of the decision tree.

2.6 Support Vector Machines

These are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. SVM maps training examples to points in space so as to maximise the width of the gap between the two categories. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. To separate the two classes of data points, there are many possible hyperplanes that could be chosen. The objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence. Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes.

2.7 Bagging

Bagging is an ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It also reduces variance and helps to avoid overfitting. Given a standard training set D of size n, bagging generates m new training sets D_i, each of size $n\prime$, by sampling from D uniformly and with replacement. By sampling with replacement, some observations may be repeated in each D_i. If $n\prime = n$, then for large n the set D_i is expected to have the fraction $(1 - \frac{1}{e})$ of the unique examples of D, the rest being duplicates. This kind of sample is known as a bootstrap sample. Sampling with replacement ensures each bootstrap is independent of its peers, as it does not depend on previous chosen samples when sampling. Then, m models are fitted using the above m bootstrap samples and combined by averaging the output.

2.8 Confusion matrix

In the field of machine learning and specifically the problem of statistical classification, a confusion matrix is a specific table layout that allows visualization of the performance of an algorithm. Each row of the matrix represents the instances in an actual class, while each column represents the instances in a predicted class. Tagged data: positive and negative are classified into a predicted positive class or a predicted negative class. It is possible that a data originally marked as positive will be mistakenly classified as negative and vice versa. For a two classes classification problem the matrix consists of 4 fields:

- True Positive – in terms of medical data – patient with some condition who were correctly diagnosed to have a condition,
- True Negative – patient without a condition who were correctly diagnosed to not have a condition,
- False Negative – patient with a condition who were incorrectly diagnosed to not have a condition,
- False Positive – patient without a condition who were incorrectly diagnosed to have a condition.

From hese values there can be calculated many derivative measures presenting the accuracy of classification and the conclusions drawn from them.

2.9 ROC curve

A receiver operating characteristic curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The ROC curve is created by plotting the true positive rate against the false positive rate at various threshold settings. In general, if the probability distributions for both detection and false alarm are known, the ROC curve can be generated by plotting the cumulative distribution function of the detection probability in the y-axis versus the cumulative distribution function of the false-alarm probability on the x-axis. Its analysis provides tools to select possibly optimal models and to discard suboptimal ones independently of the cost context or the class distribution.

3 Results

3.1 Descriptive analysis and data visualization

3.1.1 Basic characteristics

Firstly we wanted to check how the basic characteristics of our data look like. It was checked for all the variables in the set. The results such as mean and standard deviation, maximum and minimum or skewness and kurtosis are shown on Picture 1.

Statistical summary of variables

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
radius_mean	2	569	14.1272917398946	3.52404882621208	13.37	13.8199124726477	2.81694	6.981	28.11	21.129	0.937416783784186	0.814141774990952	0.147735812093463
texture_mean	3	569	19.2896485061511	4.30103576816695	18.84	19.0377899343545	4.166106	9.71	39.28	29.57	0.647024124971888	0.728007079464299	0.180308799165014
perimeter_mean	4	569	91.9690333919156	24.2989810387549	86.24	89.7404595185996	18.843846	43.79	188.5	144.71	0.985433432069862	0.939282126825769	1.01866627672771
area_mean	5	569	654.889103690685	351.914129181653	551.1	606.127789934354	227.28258	143.5	2501	2357.5	1.63706537091489	3.58654877611044	14.7530077549177
smoothness_mean	6	569	0.0963602811950791	0.0140641281376736	0.09587	0.0958773960612691	0.0140847	0.05263	0.1634	0.11077	0.453920658765969	0.824467062691694	0.000589598922793331
compactness_mean	7	569	0.104340984182777	0.0528127579325122	0.09263	0.09808295404814	0.048377238	0.01938	0.3454	0.32602	1.18385556918689	1.6088967210702	0.00221402598738728
concavity_mean	8	569	0.0887993158172232	0.0797198087078935	0.06154	0.0772507002188184	0.059985996	0	0.4268	0.4268	1.39380080370043	1.95313568032897	0.00334202823519206
concave.points_mean	9	569	0.0489191458699473	0.0388028448591536	0.0335	0.0440001312910284	0.029859564	0	0.2012	0.2012	1.16501237716024	1.03246889085676	0.00162669987832307
symmetry_mean	10	569	0.181161862917399	0.0274142813360357	0.1792	0.179522975929978	0.02535246	0.106	0.304	0.198	0.721787749376178	1.25113501270688	0.00114926645908344
*fractal_dimension_*	11	569	0.0627976098418278	0.00706036279508446	0.06154	0.062078533916849	0.006256572	0.04996	0.09744	0.04748	1.297619074017	2.94805460754254	0.000295985805715254
radius_se	12	569	0.405172056239016	0.277312732986104	0.3242	0.357696717724289	0.1571556	0.1115	2.873	2.7615	3.07234682329623	17.4490949504915	0.0116255545345541
texture_se	13	569	1.21685342706503	0.551648392617202	1.108	1.15584573304158	0.46746378	0.3602	4.885	4.5248	1.63777325641993	5.26263349768412	0.02312630366162
perimeter_se	14	569	2.86605922671353	2.02185455404211	2.287	2.51091466083151	1.141602	0.757	21.98	21.223	3.42548033134961	21.11877478893	0.0847605521962488
area_se	15	569	40.337079086116	45.4910055161318	24.53	31.6904157549234	13.625094	6.802	542.2	535.398	5.41850014723589	48.5853969954483	1.90708215870489
smoothness_se	16	569	0.00704097891036907	0.00300251794383907	0.00638	0.00664756455142232	0.0021512526	0.001713	0.03113	0.029417	2.30226162953601	10.3205924127678	0.000125872100141999
compactness_se	17	569	0.0254781388400703	0.0179081793256774	0.02045	0.0226609781181619	0.012987576	0.002252	0.1354	0.133148	1.8922031772299	5.02269223928606	0.000750749931759063
concavity_se	18	569	0.031893716344464	0.0301860603229884	0.02589	0.0278033435448578	0.018502848	0	0.396	0.396	5.08355017350084	48.2419737785116	0.00126546547895378
concave.points_se	19	569	0.0117961370826011	0.00617028517404687	0.01093	0.0112518512035011	0.005166861	0	0.05279	0.05279	1.43707013682529	5.04249615681978	0.000258671810746703
symmetry_se	20	569	0.0205422987697715	0.0082663715287984	0.01873	0.0193385557986871	0.005826618	0.007882	0.07895	0.071068	2.18357282407941	7.77840246277522	0.000346544321914518
fractal_dimension_se	21	569	0.00379490386643234	0.0026460709670892	0.003187	0.00336937199124726	0.0015923124	0.0008948	0.02984	0.0289452	3.9033040976398	25.9379658269062	0.000110929065531719
radius_worst	22	569	16.2691898066784	4.83324158046932	14.97	15.730590809628	3.647196	7.93	36.04	28.11	1.09730594719807	0.911502584161864	0.202620027459165
texture_worst	23	569	25.677223198594	6.14625762303832	25.41	25.3909628008753	6.419658	12.02	49.54	37.52	0.495697034689782	0.200529985940285	0.25766452340878
perimeter_worst	24	569	107.261212653779	33.6025422690364	97.66	103.424814004376	25.011462	50.41	251.2	200.79	1.12222270056434	1.03601883992582	1.40869185284729
area_worst	25	569	880.583128295255	569.356992669949	686.5	788.019912472648	319.64856	185.2	4254	4068.8	1.84958138450899	4.32152808249216	23.8686867950124
smoothness_worst	26	569	0.132368594024605	0.0228324294048355	0.1313	0.131559080962801	0.02179422	0.07117	0.2226	0.15143	0.413238267298243	0.490458581048055	0.000957185233955963
compactness_worst	27	569	0.254265043936731	0.157336488913742	0.2119	0.234251356673961	0.12913446	0.02729	1.058	1.03071	1.46579482081107	2.98104166391894	0.00659588873704415
concavity_worst	28	569	0.272188483304042	0.208624280608132	0.2267	0.248481159737418	0.1957032	0	1.252	1.252	1.14417940990005	1.57444668578961	0.00874598481406007
concave.points_worst	29	569	0.114606223198594	0.0657323411959421	0.09993	0.11096249452954	0.066079482	0	0.291	0.291	0.490021300053555	-0.550001419557813	0.00275564309301165
symmetry_worst	30	569	0.290075571177504	0.0618674675375187	0.2822	0.284189715536105	0.05070492	0.1565	0.6638	0.5073	1.42637637129183	4.36910290715884	0.00259361916067597
fractal_dimension_worst	31	569	0.0839458172231986	0.018061267348894	0.08004	0.0817631291028446	0.014618436	0.05504	0.2075	0.15246	1.65382374102586	5.15935586551946	0.000757167715548965

Figure 1: Basic characteristics

3.1.2 Properties of distributions

Now for each of the 10 features we introduce charts of the density function and a box plots for the mean, standard error and the worst parameters.

First category is the tumor radius shown on the Pictures 2 and 3. We can see that the average of the tumor is around 14 mm. Nevertheless, the distribution is right skewed, so there exist a lot of data of bigger values. Half of the data between first and third quartile is distributed between 12 and 16 mm. The standard error is relatively small, but it has some outlier values.

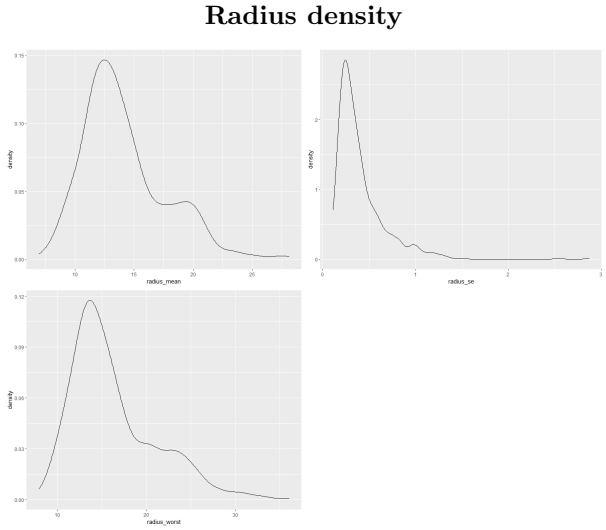

Figure 2: Density functions for the radius of tumor

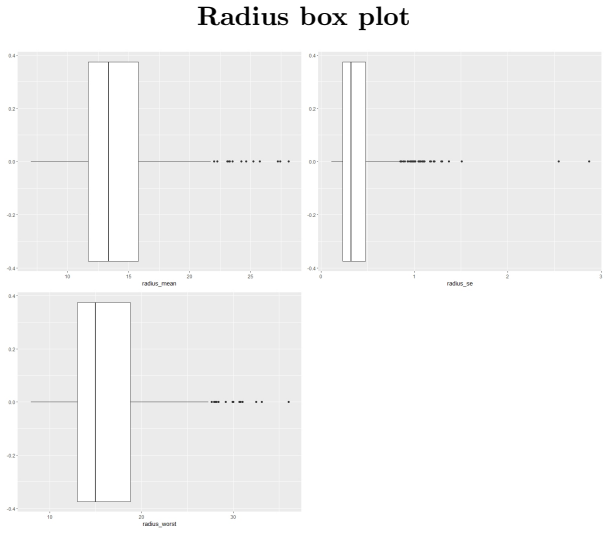

Figure 3: Box plots for the radius of tumor

Next category is the tumor texture (standard deviation of gray-scale values) shown on the Pictures 4 and 5. The density function with its mean around 19 mm is also right skewed, but more close to normal distribution. The standard error is much bigger comparing to the radius one.

Texture density

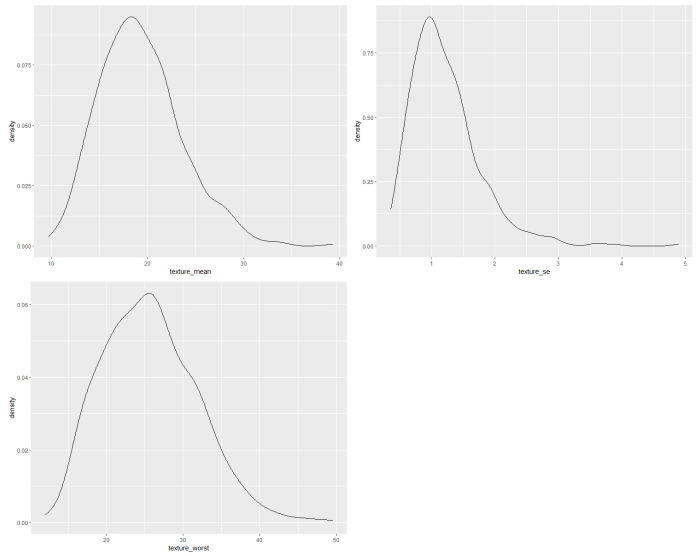

Figure 4: Density functions for the texture of tumor

Texture box plot

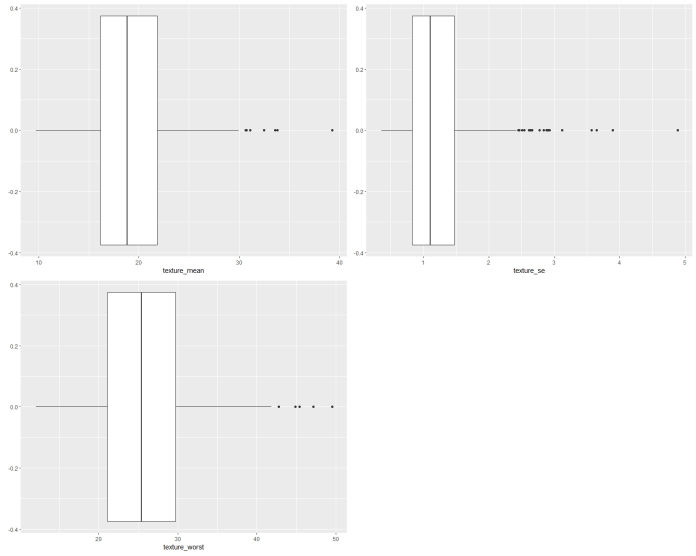

Figure 5: Box plots for the texture of tumor

Now let us see the category of the tumor perimeter shown on the Pictures 6 and 7. The density function with its mean around 92 mm is also right skewed, similarly to radius density shape. The standard error is also similar to the radius one.

Perimeter density

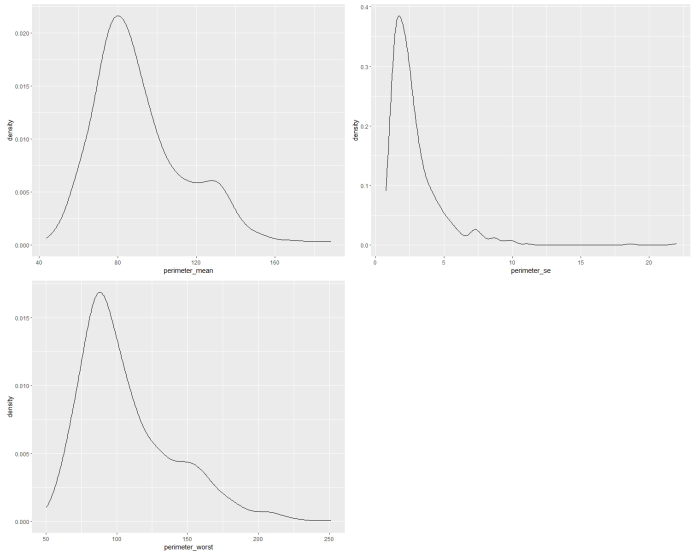

Figure 6: Density functions for the perimeter of tumor

Perimeter box plot

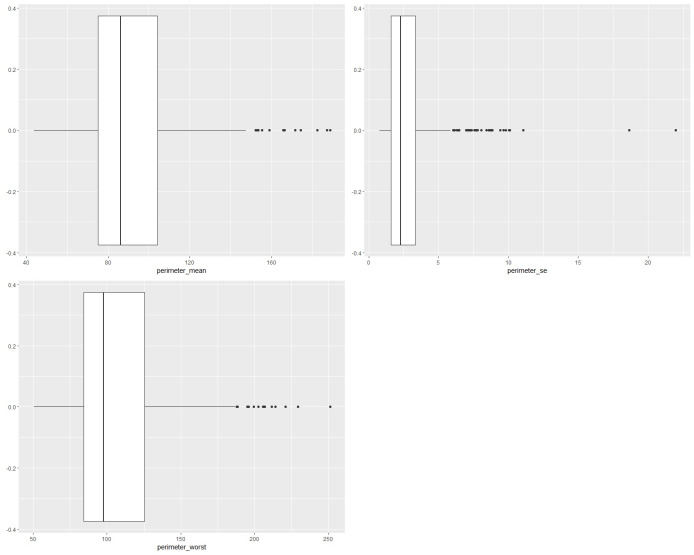

Figure 7: Box plots for the perimeter of tumor

Next category is the area of tumor shown on the Pictures 8 and 9. The mean area of tumor is around 655mm^2 and the density function is also right skewed, similarly to radius and perimeter density shape. The standard error in this case is the biggest of all categories with it mean value of almost 15. There are a lot of outliers here, what has big impact on the results.

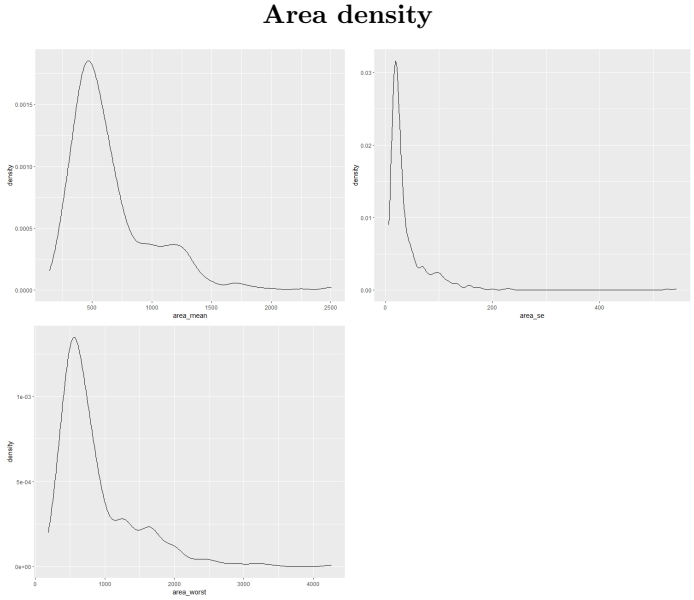

Figure 8: Density functions for the area of tumor

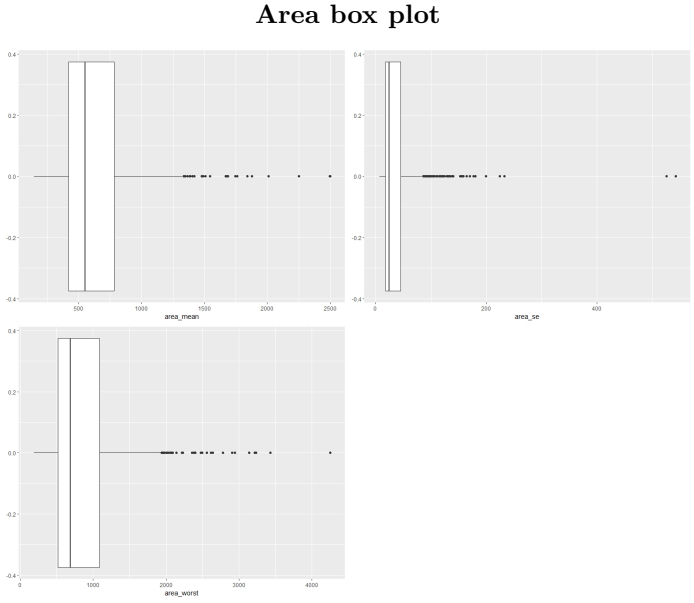

Figure 9: Box plots for the area of tumor

Now let us see the smoothness (local variation in radius lengths) of the tumor shown on the Pictures 10 and 11. The shape of the density function of mean is similar to normal distribution. Here the standard error is really small, but it may be caused by the small values of the feature.

Smoothness density

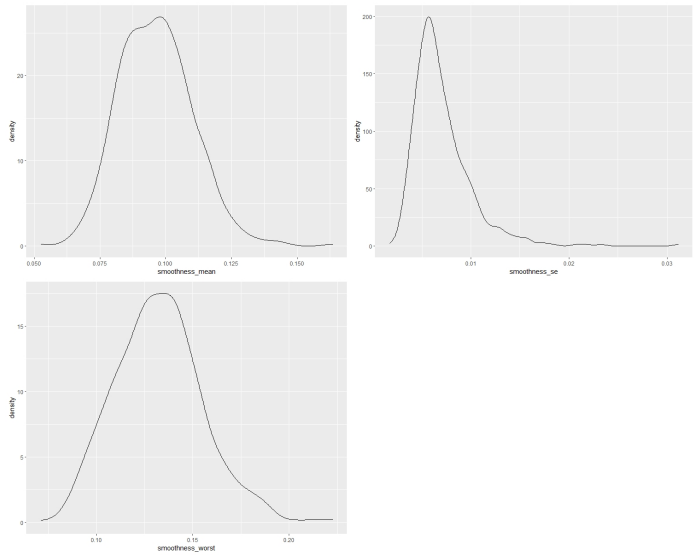

Figure 10: Density functions for the smoothness of tumor

Smoothness box plot

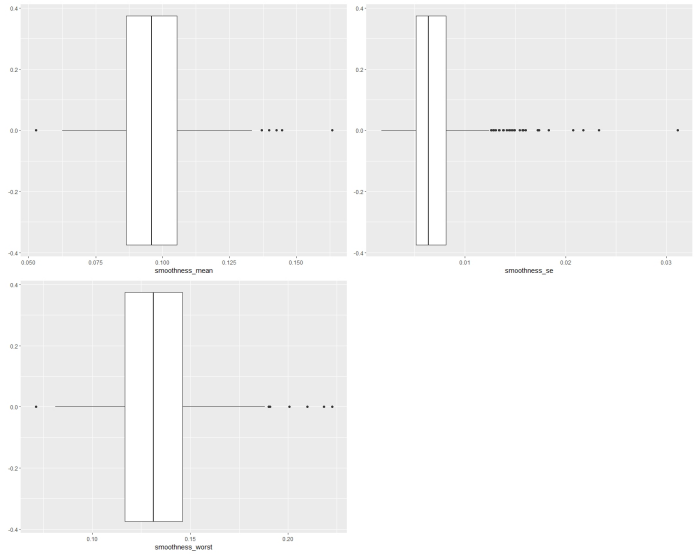

Figure 11: Box plots for the smoothness of tumor

Now let us see the compactness ($\frac{\text{perimeter}^2}{\text{area}} - 1$) of the tumor shown on the Pictures 12 and 13. The denisty functions are also right skewed. The mean of the worst cases are more than twice bigger than for all the data.

Compactness density

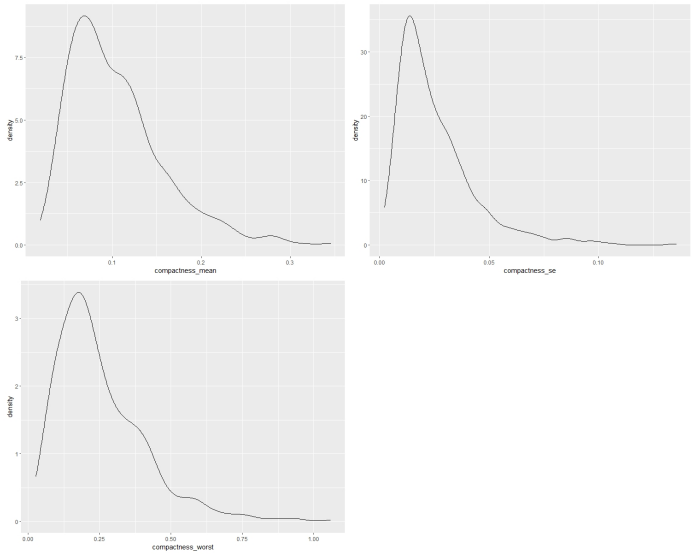

Figure 12: Density functions for the compactness of tumor

Compactness box plot

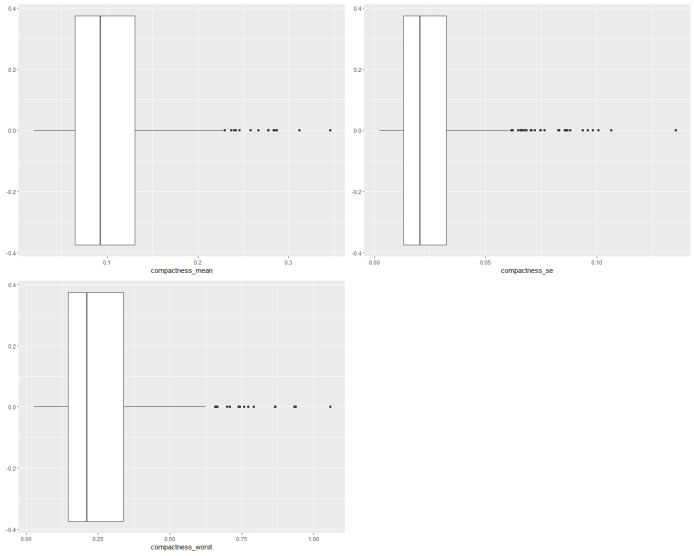

Figure 13: Box plots for the compactness of tumor

Next category is the concavity (severity of concave portions of the contour) of the tumor shown on the Pictures 14 and 15. The shape of the density function of the mean is a bit different, but the shape is also right skewed. The mean of the worst cases is significantly bigger than the mean of all observations.

Concavity density

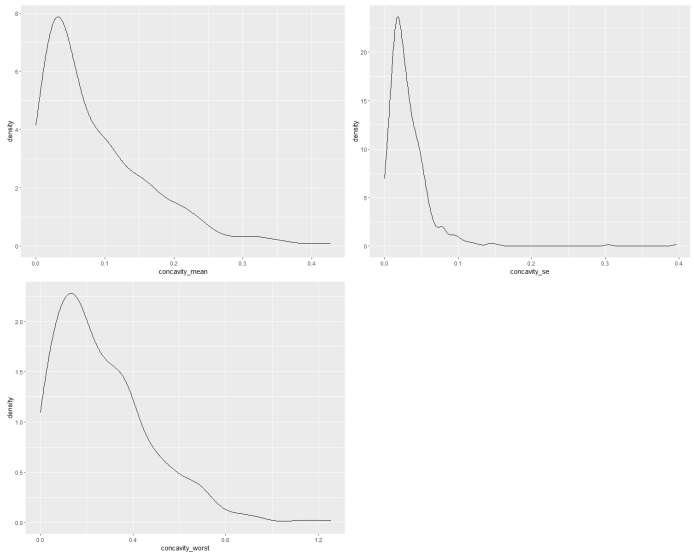

Figure 14: Density functions for the concavity of tumor

Concavity box plot

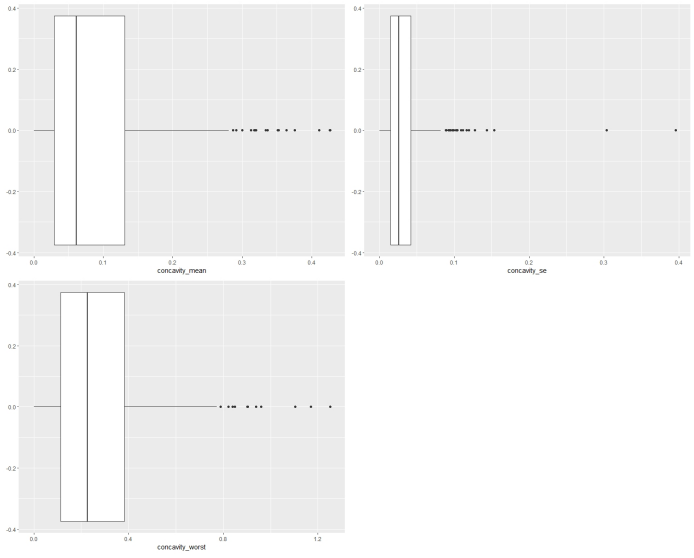

Figure 15: Box plots for the concavity of tumor

Let us see the concave points (number of concave portions of the contour) of the tumor shown on the Pictures 16 and 17. The shape of the density function of the mean is also right skewed. We cann see a bit different behaviour of the standard error, meaning that the mean is bigger and shifted to the right. For the worst cases the mean is also significantly bigger.

Concave points density

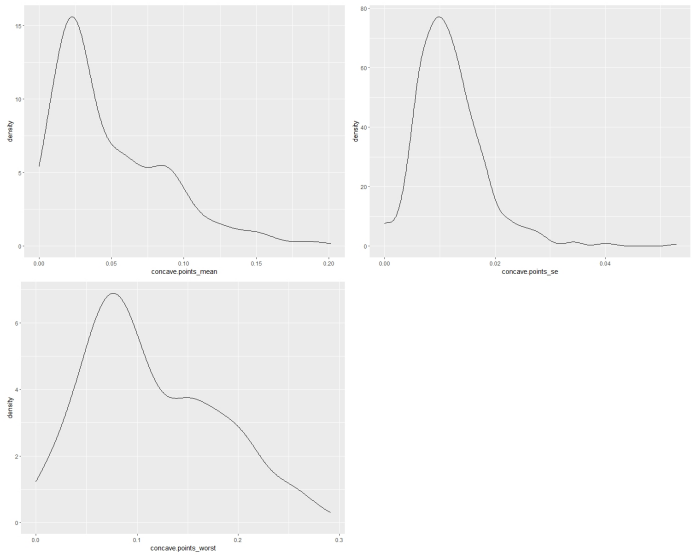

Figure 16: Density functions for the concave points of tumor

Concave points box plot

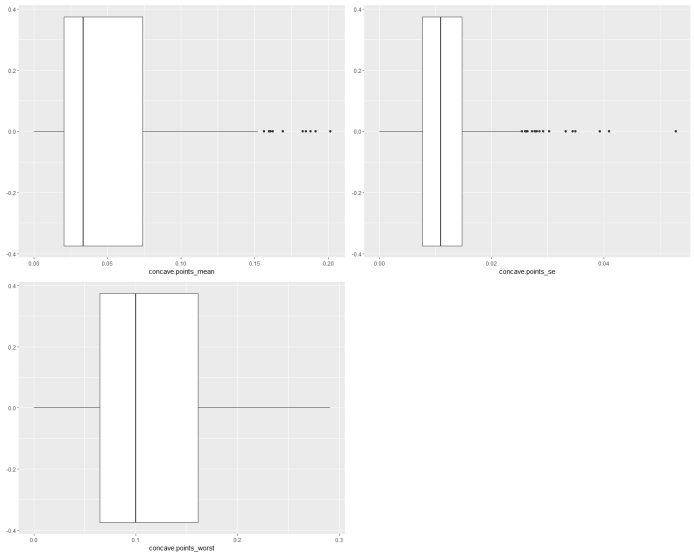

Figure 17: Box plots for the concave points of tumor

Next category is the symmetry of the tumor shown on the Pictures 18 and 19. The shape of the mean density is close to the normal distribution, so there are less outliers.

Symmetry density

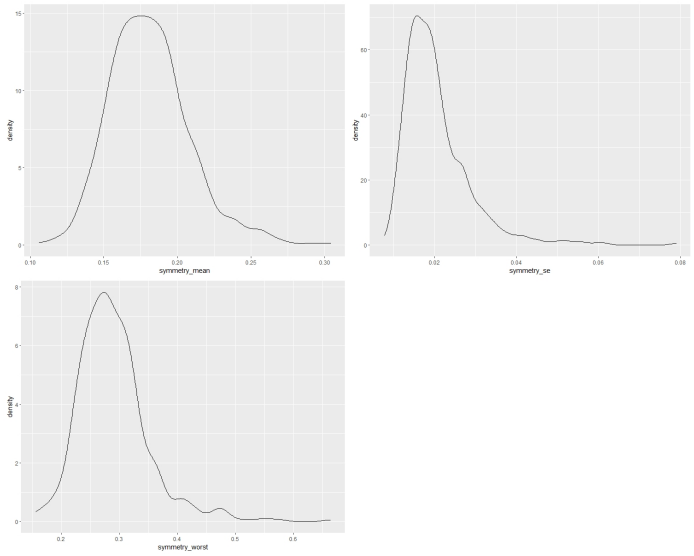

Figure 18: Density functions for the symmetry of tumor

Symmetry box plot

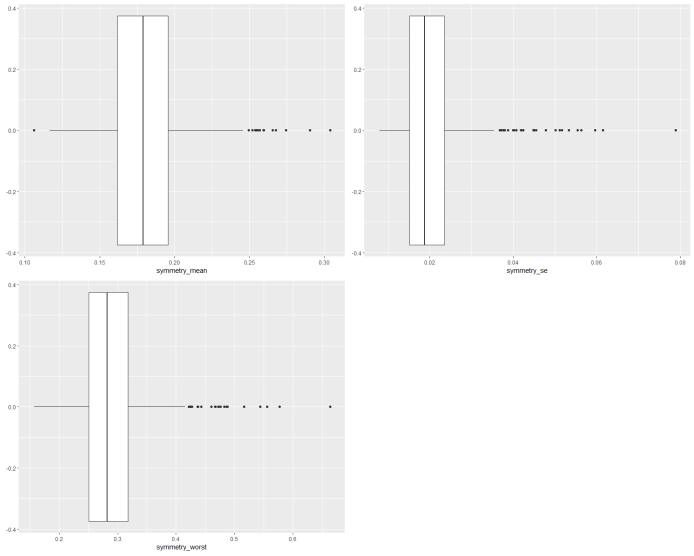

Figure 19: Box plots for the symmetry of tumor

Next category is the fractal dimension ("coastline approximation" - 1) of the tumor shown on the Pictures 20 and 21. The density function of mean is rigt skewed. Here the mean of the worst cases are quite similar to the all data.

Fractal dimension density

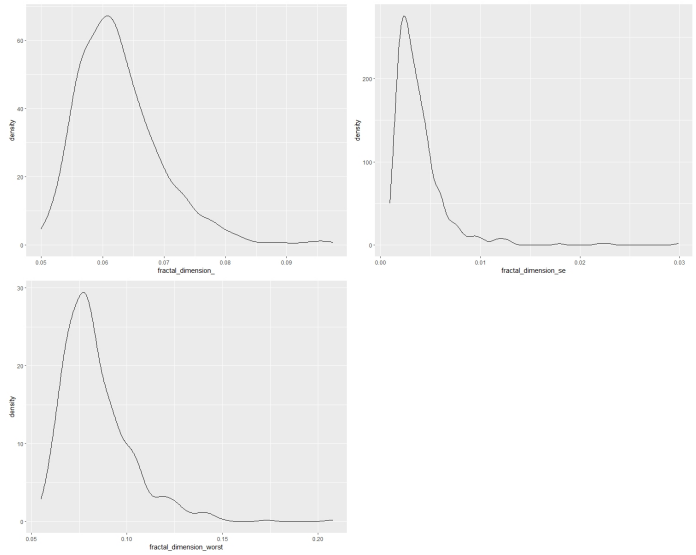

Figure 20: Density functions for the fractal dimension of tumor

Fractal dimension box plot

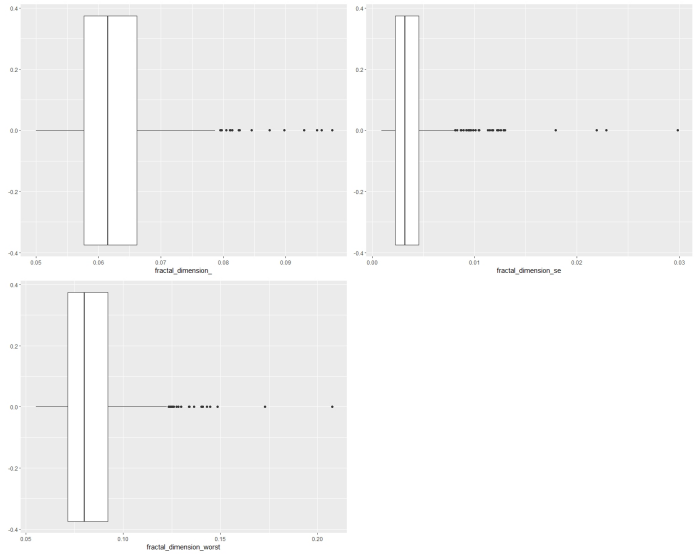

Figure 21: Box plots for the fractal dimension of tumor

3.1.3 Scatter plots

We wanted to visualize data in order to some features. On the Figures 22, 23 and 24 we can see the scatter plots with respect to other features.

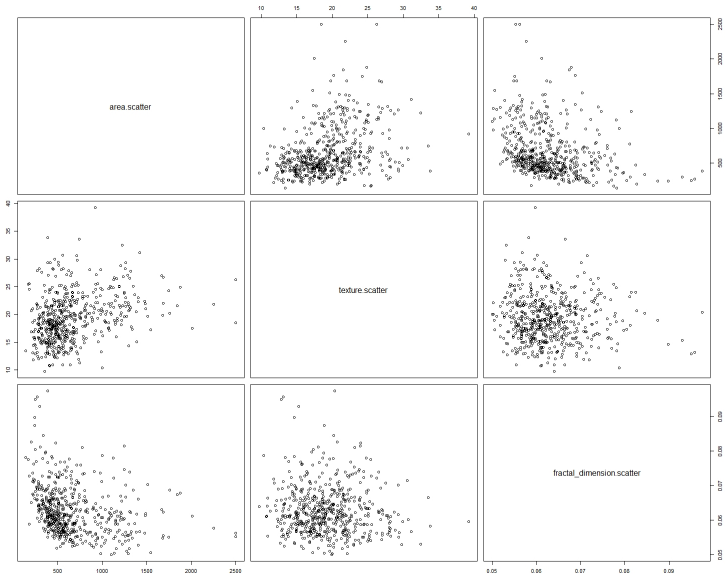

Figure 22: Scatter plot with respect to area, texture and fractal dimension

As we can see on the Figure 22, there is no big correlation between these three features.

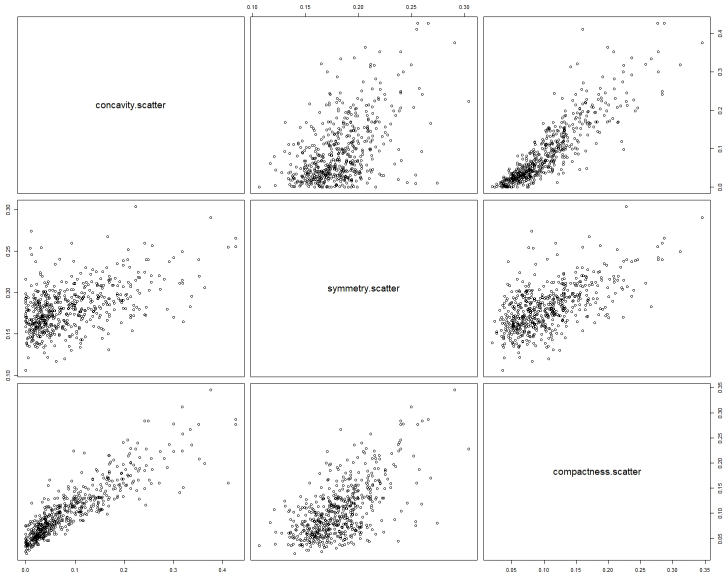

Figure 23: Scatter plot with respect to concavity, symmetry and compactness

As we can see on the Figure 23, these features are significantly more correlated than the features before. Concavity and compactness seem to be correlated.

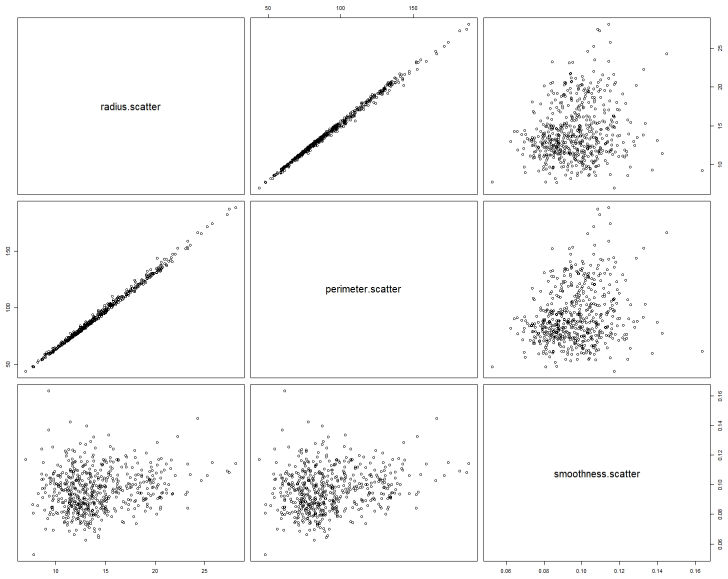

Figure 24: Scatter plot with respect to radius, perimeter and smoothness

As we can see on the Figure 24, there is high correlation between radius and perimeter.

3.1.4 Correlation

In the section before we could see that some of our features are more or less correlated with each others. Let us check how it looks like for our data. The results are shown on the Figure 25.

Correlation matrix

	symmetry_mean	smoothness_mean	fractal_dimension_	texture_mean	area_mean	radius_mean	perimeter_mean	compactness_mean	concavity_mean	concave.points_mean
symmetry_mean	1	0.56	0.48	0.071	0.15	0.15	0.18	0.6	0.5	0.46
smoothness_mean	0.56	1	0.58	-0.023	0.18	0.17	0.21	0.66	0.52	0.55
*fractal_dimension_*	0.48	0.58	1	-0.076	-0.28	-0.31	-0.26	0.57	0.34	0.17
texture_mean	0.071	-0.023	-0.076	1	0.32	0.32	0.33	0.24	0.3	0.29
area_mean	0.15	0.18	-0.28	0.32	1	0.99	0.99	0.5	0.69	0.82
radius_mean	0.15	0.17	-0.31	0.32	0.99	1	1	0.51	0.68	0.82
perimeter_mean	0.18	0.21	-0.26	0.33	0.99	1	1	0.56	0.72	0.85
compactness_mean	0.6	0.66	0.57	0.24	0.5	0.51	0.56	1	0.88	0.83
concavity_mean	0.5	0.52	0.34	0.3	0.69	0.68	0.72	0.88	1	0.92
concave.points_mean	0.46	0.55	0.17	0.29	0.82	0.82	0.85	0.83	0.92	1

Figure 25: Correlation matrix

From the table on Figure 25 we can see, that there are some features highly correlated with others. For example radius with area (so the bigger the radius of the tumor, the bigger its area), perimeter with area (so the bigger the perimeter of the tumor, the bigger its area), concavity with concave points (so the bigger the severity of the concave portions of the countour, the bigger its the number). To visualize the strength of correlation and better understanding the values, let us introduce the heat map of correlation on Figure 26.

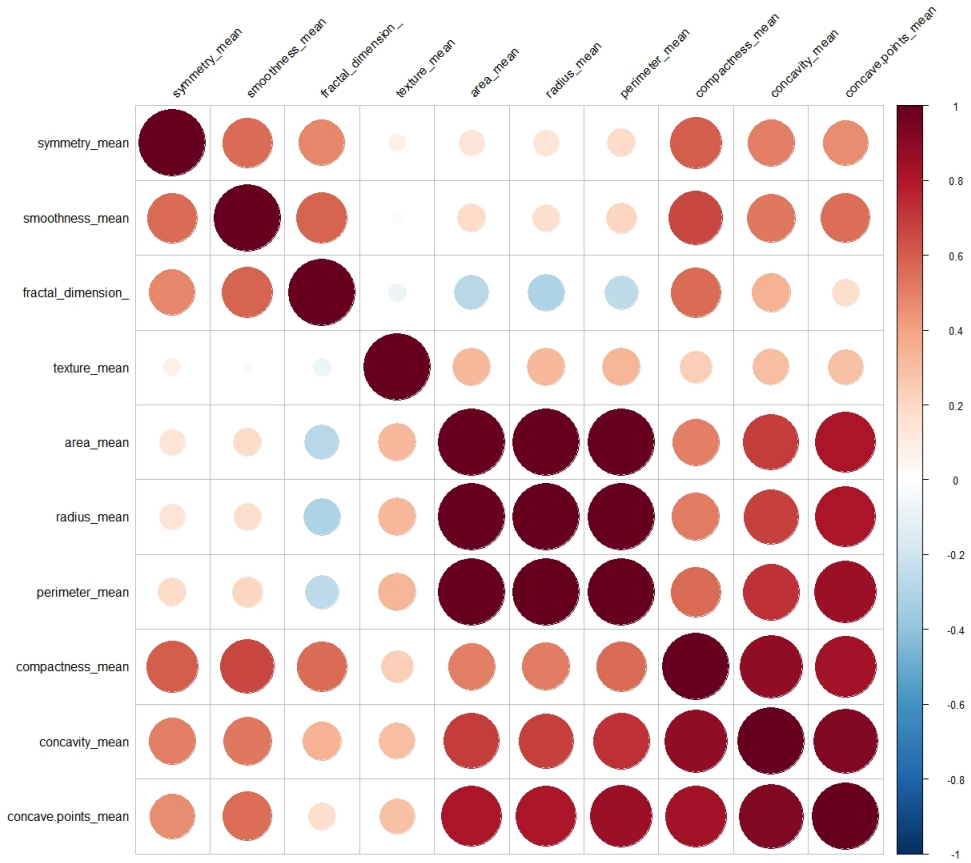

Figure 26: Heat map of correlation

3.1.5 Outliers

In our analysis we wanted to check if we have any outliers in our dataset for each variable. We decided to see the last 4 values in each feature and check if it is an outlier. The results are shown on the Figure 27.

	observation number	is_outlier
radius1	213	1
radius2	462	0
radius3	181	0
radius4	353	0
texture1	240	1
texture2	233	0
texture3	260	0
texture4	220	0
perimeter1	213	1
perimeter2	462	1
perimeter3	181	0
perimeter4	353	0
area1	462	1
area2	213	1
area3	181	1
area4	353	1
smoothness1	505	1
smoothness2	123	0
smoothness3	4	0
smoothness4	106	0
compactness1	79	1
compactness2	259	1
compactness3	123	0
compactness4	4	0
concavity1	123	1
concavity2	109	1
concavity3	153	1
concavity4	79	0
concave.points1	123	1
concave.points2	353	0
concave.points3	181	0
concave.points4	83	0
symmetry1	26	1
symmetry2	79	1
symmetry3	61	0
symmetry4	147	0
fractal_dimension1	4	1
fractal_dimension2	506	1
fractal_dimension3	505	1
fractal_dimension4	153	1

Figure 27: Outliers

If the number was 1, it means it is an outlier. So wanted to see how many of them we have for those observations, so we summed them up. The results are shown on Figure 28.

	observation number	times being an outlier
1	4	1
2	26	1
3	79	2
4	109	1
5	123	2
6	153	2
7	181	1
8	213	3
9	240	1
10	259	1
11	353	1
12	462	2
13	505	2
14	506	1

Figure 28: Outliers summary

Clearly, if the number was bigger than 1 (and we have values of 2 and 3), it could mean those observations are outliers.

3.1.6 Diagnosis categories

For better understanding of the data we needed to divide our data into some categories and than clear some of the features that are not crucial. Basically, it was the mean of each variable that is useful for our analysis. What is more, very important part for our analysis is categorical attributes that are benign and malignant tumor. You can see the pie chart of number of each on Figure 29

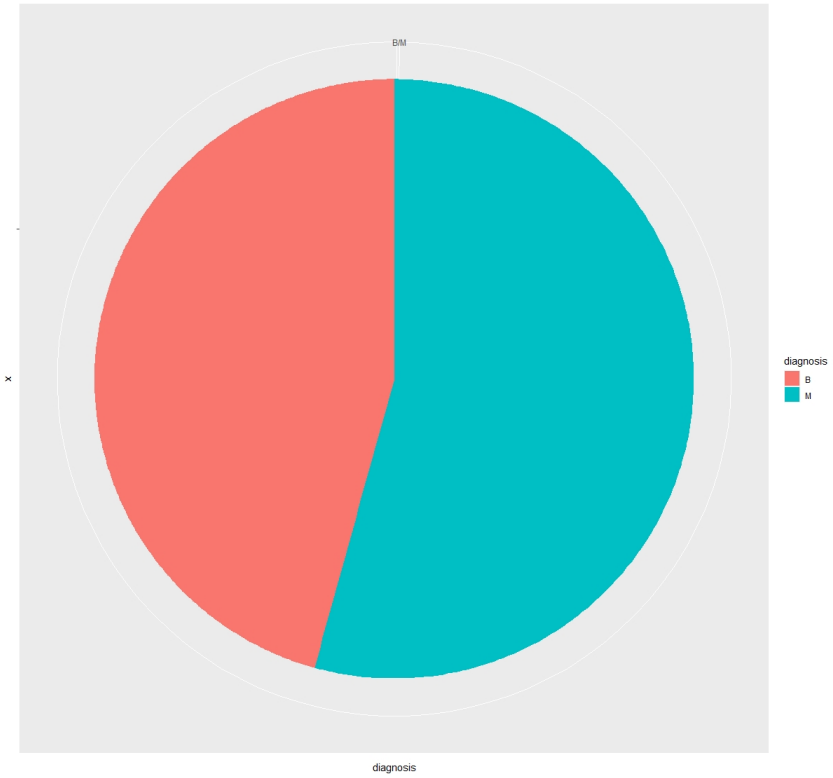

Figure 29: Pie chart of diagnosis categories

We wanted to present the differences for all the features in order to these categories. Basic characteristics are shown on the Pictures 30 and 31. For almost all features, the mean values are bigger for the malignant tumor than the benign tumor. The same behaviour we can see for other characteristics such as median, minimum or maximum or standard error. What is interesting, for the last feature, fractal dimension, all the characteristics are bigger for malignant tumor than for the benign tumor.

What is more, we can now see the differences in the skewness and curtosis. For malignant tumor, skewness for all variables are bigger than 0, meaning the distribution is right skewed. For benign tumor we can see two values (for radius and perimeter) than are smaller than 0, what could mean there are left skewed, but actually the values are really close to 0, so we can say it is symmetrical distribution. In case of curtosis, for malignant tumor all values are positive, so the distribution is leptokurtic. For benign tumor, we can see negative values for the same features as before, but still close to 0. What is more, there are some values that are extraordinarily big (for fractal dimension and concavity points).

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
radius_mean	1	357	12.1465238095238	1.78051164614104	12.2	12.1689721254355	1.690164	6.981	17.85	10.869	-0.0830962316590464	-0.0454932166029534	0.0942346692617913
texture_mean	2	357	17.9147619047619	3.99512459367591	17.39	17.5159581881533	3.469284	9.71	33.81	24.1	0.973112637329284	1.16245842116146	0.211444415744572
perimeter_mean	3	357	78.075406162465	11.8074375801087	78.18	78.1621602787456	11.134326	43.79	114.6	70.81	-0.0641414893185121	-0.0495921545791025	0.624915864831504
area_mean	4	357	462.790196078431	134.28711814706	458.4	459.4	127.05882	143.5	992.1	848.6	0.339394632921947	0.27002333246203	7.10722796569951
smoothness_mean	5	357	0.0924776470588235	0.0134460773302741	0.09076	0.0919115679442509	0.013254444	0.05263	0.1634	0.11077	0.657883040336225	1.78582699524552	0.000711641877116081
compactness_mean	6	357	0.0800846218487395	0.0337499545931426	0.07529	0.0766972473867596	0.031609032	0.01938	0.2239	0.20452	1.20227301015476	2.20855818235554	0.00178623701539851
concavity_mean	7	357	0.0460576210084034	0.0434421510450616	0.03709	0.0401753449477352	0.028302834	0	0.4108	0.4108	3.44413149679693	20.3957674857682	0.00229920244814163
concave.points_mean	8	357	0.025717406162465	0.0159087783782748	0.02344	0.0243150348432056	0.013002402	0	0.08534	0.08534	0.917207018658975	0.984351382741467	0.000841981838245785
symmetry_mean	9	357	0.174185994397759	0.0248067582067933	0.1714	0.173022299651568	0.02357334	0.106	0.2743	0.1683	0.653416042409648	1.24707916617143	0.00131291286981519
fractal_dimension_	10	357	0.0628673949579832	0.00674734281392506	0.06154	0.0620685017421603	0.005530098	0.05185	0.09575	0.0439	1.63681816465515	4.36377214129747	0.000357107250516567

Figure 30: Basic characteristics for benign tumor

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
radius_mean	1	212	17.4628301886792	3.20397110077937	17.325	17.3221764705882	3.358089	10.95	28.11	17.16	0.494821970611073	0.306035121320336	0.220049638639178
texture_mean	2	212	21.6049056603774	3.77946992077634	21.46	21.4342941176471	3.246894	10.38	39.28	28.9	0.693618140644327	2.24536119825195	0.259575059872473
perimeter_mean	3	212	115.365377358491	21.8546532910737	114.2	114.194941176471	23.173038	71.9	188.5	116.6	0.596397169800873	0.515803616211467	1.50098375048248
area_mean	4	212	978.37641509434	367.93797760667	932	945.975294117647	366.57285	361.6	2501	2139.4	1.10190185173221	2.17467814280074	25.2700840510962
smoothness_mean	5	212	0.102898490566038	0.0126082355010583	0.1022	0.102422294117647	0.012401949	0.07371	0.1447	0.07099	0.472983721382877	0.361703632018342	0.000865937169411627
compactness_mean	6	212	0.145187783018868	0.0539874950527983	0.13235	0.141001058823529	0.04462626	0.04605	0.3454	0.29935	0.824964255764145	0.771151987977071	0.00370787638331472
concavity_mean	7	212	0.160774716981132	0.0750193278502571	0.15135	0.154541176470588	0.06775482	0.02398	0.4268	0.40282	0.888909364069228	1.05899555622896	0.00515234859028145
concave.points_mean	8	212	0.08799	0.0343739088754015	0.08628	0.0854822352941177	0.030267279	0.02031	0.2012	0.18089	0.728665781270571	0.650138512310603	0.002360809754123
symmetry_mean	9	212	0.192908962264151	0.0276380921430302	0.1899	0.19098	0.02646441	0.1308	0.304	0.1732	0.800506085068288	1.21006487843762	0.00189819196161623
fractal_dimension_	10	212	0.0626800943396226	0.00757331502480859	0.061575	0.0620891176470588	0.00763538999999999	0.04996	0.09744	0.04748	0.882144398299384	1.22686711622277	0.000520137411384394

Figure 31: Basic characteristics for malignant tumor

We can also show on figure 32 how much the values are bigger for Malignant tumor than for Benign, all cells ih this table are the result of division the values from the tables 31 and 30.

	mean	sd	max	range
radius_mean	1.43768130392887	1.79946652285226	1.57478991596639	1.57880209770908
texture_mean	1.20598341050989	0.94602053882351	1.16178645371192	1.19917012448133
perimeter_mean	1.47761482173311	1.85092261913719	1.64485165794066	1.64666007626042
area_mean	2.11408198225645	2.7399350189624	2.52091523031952	2.5210935658732
smoothness_mean	1.11268499836058	0.93768875422653	0.885556915544676	0.640877493906292
compactness_mean	1.81292961953535	1.59963163517167	1.54265297007593	1.46367103461764
concavity_mean	3.4907299478581	1.72687875820056	1.03894839337877	0.980574488802337
concave.points_mean	3.42141814163292	2.16068814701341	2.35762831028826	2.11963909069604
symmetry_mean	1.10748836570429	1.11413558807783	1.10827561064528	1.0291146761735
fractal_dimension_	0.997020703363234	1.12241444279056	1.0176501305483	1.08154897494305

Figure 32: The average change in value between the two types of diagnosis

3.1.7 Properties of distribution for diagnosis categories

As we could see the differences in the section before, we wanted to check the properties of the distributions in order to diagnosis categories. First comparision is shown on the Figure 33.

Box plots in order to diagnosis categories

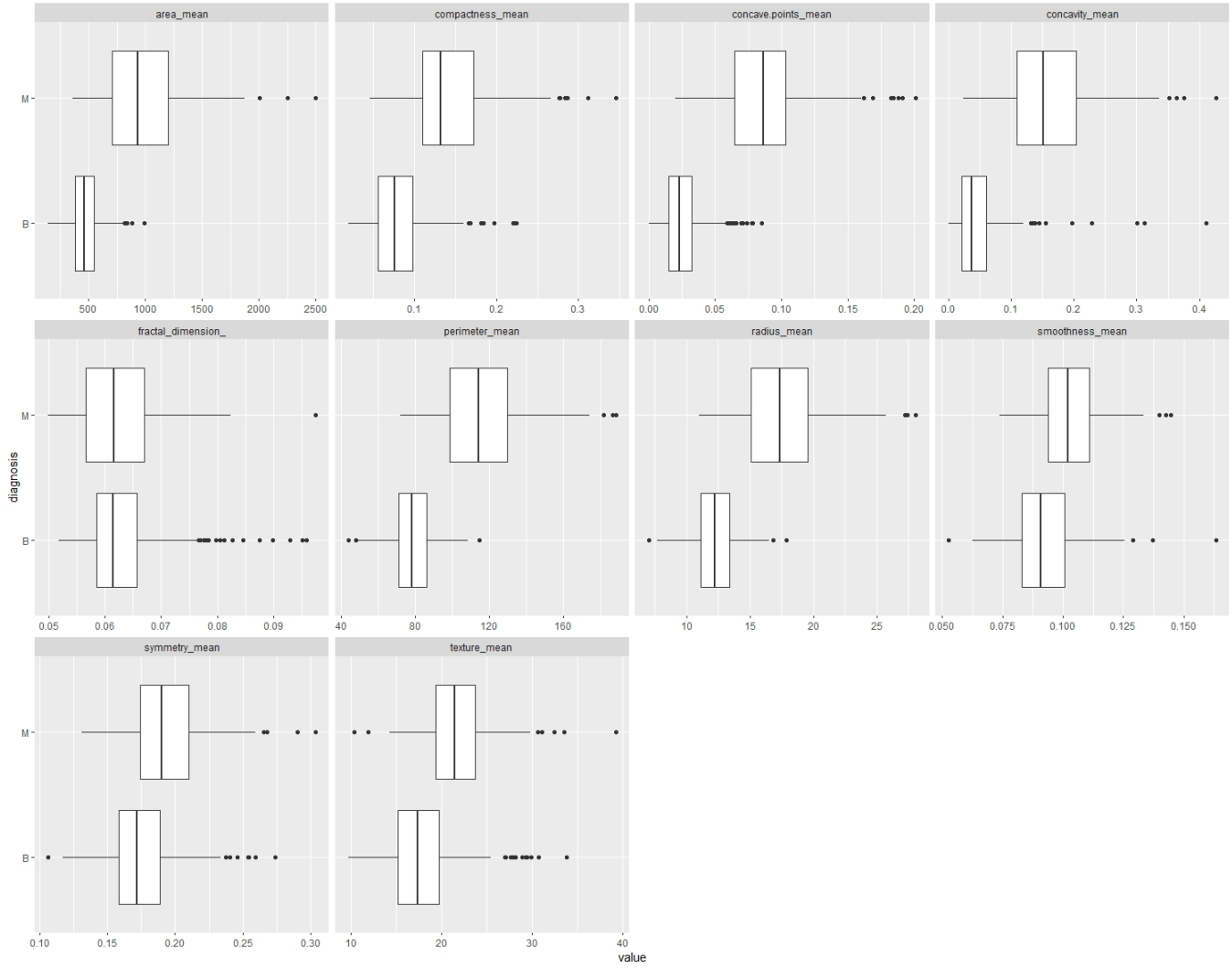

Figure 33: Box plots in order to diagnosis categories for each feature

As we can see, for the malignant tumor for almost all features the box plots are visibly shifted to the right, meaning its values are more bigger than for the benign tumor. What is more, the range of the values is also much bigger. Only for the fractal dimension the results are quite similar.

To see more differences in distributions, we wanted also to compare the density functions for each feature in order to diagnosis categories. The comparision are shown on the Figures 34, 35 and 36.

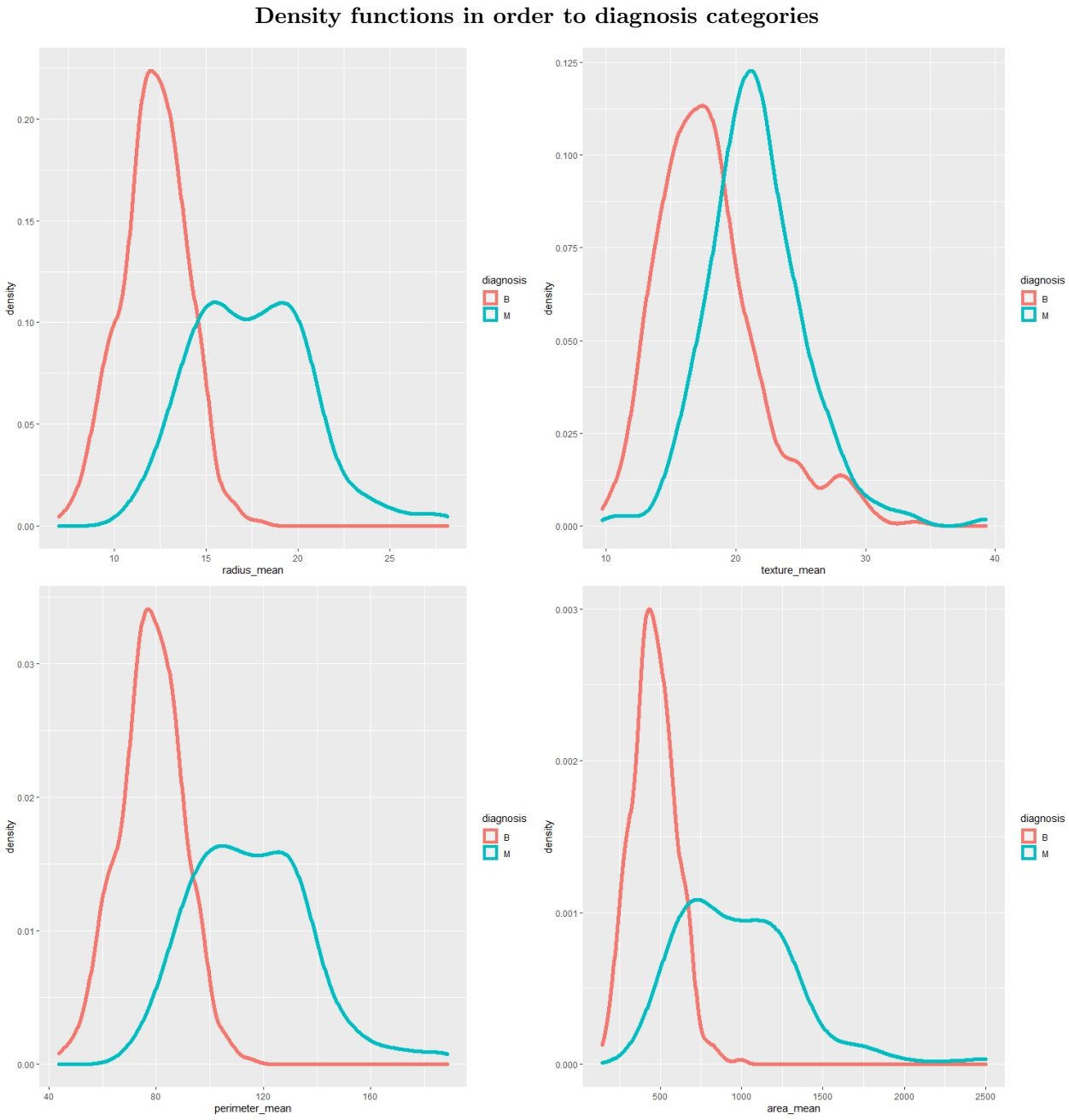

Figure 34: Density functions in order to diagnosis categories for radius, texture, perimeter and area

Density functions in order to diagnosis categories

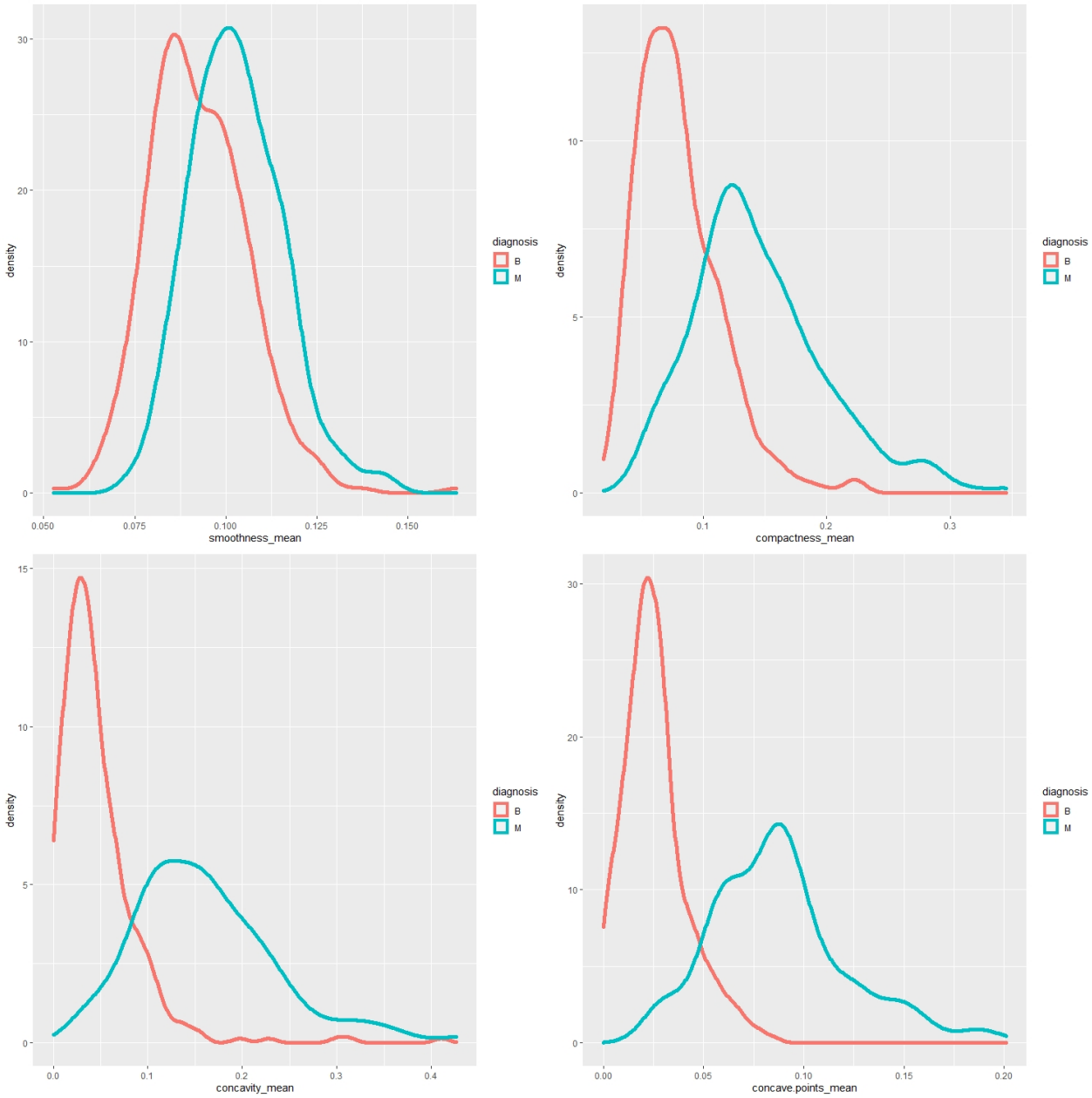

Figure 35: Density functions in order to diagnosis categories for smoothness, compactness, concavity and concave points

Density functions in order to diagnosis categories

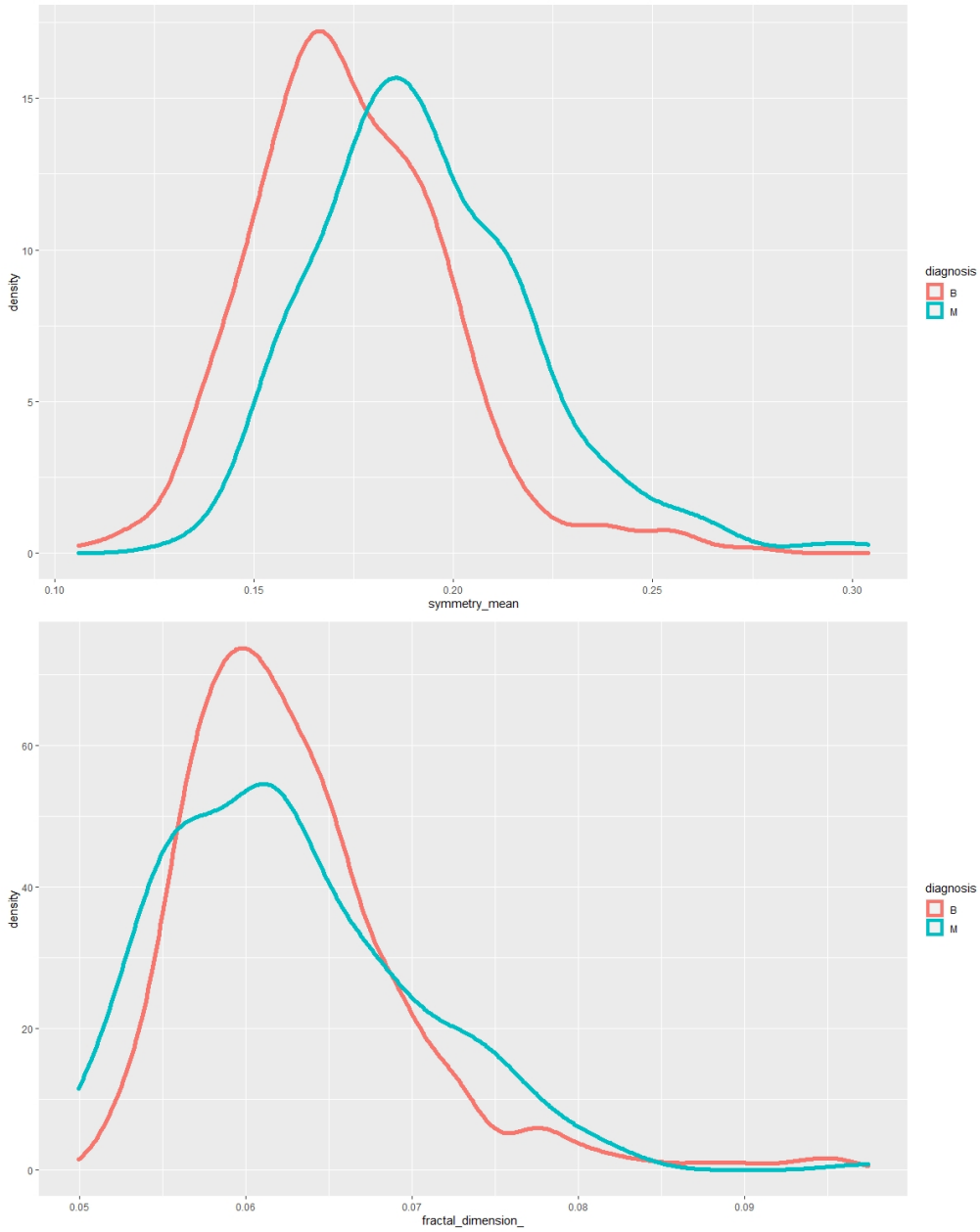

Figure 36: Density functions in order to diagnosis categories for symmetry and fractal dimension

There are no doubts the density plots are significantly different in order to diagnosis category. We can clearly see that the mean of each is shifted to the right, meaning there are more bigger values.

3.2 Classification

There are many ways to classify our data, but in this project we would like to focus on checking whether the diagnosis of the tumor was correctly assigned, as for the medical information, it is really essential to be sure that the patient was given a precise recognition of his disease. There are many methods which we will utilize that will help us classify the tumor according to the variables used in particular models. The main issue would be to check the predictions of these methods, determine their accuracy for various combinations of features and pick the one for which the misclassification error is the smallest. In addition we would like to minimize the *false negative* error which would indicate that the patient's tumor was predicted to be benign while in fact it was malignant – the worst scenario for a medical case. Those two criteria will result in choosing the best method for these type of analysis and will help us compare different models and approaches to this assignment. In order to accomplish that, we randomly divided the data into two sets – training set and test set, with the proportion of 70% being allocated to the training one. For the results to be meaningful, we have selected only the *mean* values of the features and leave the variables that were described as *se* and *worst* as they were the derivatives of the main category. We have also decided to create three different feature subsets that will be used for each of the methods, they are characterised as follows:

1. The first one contains all the *mean* variables as the predictors for the formula for the model to be created – marked as *all* ,
2. For the next one we've decided to take into the consideration the values from the table 32 and the correlation table 25. We wanted the predictors to be much bigger when the diagnosis is malignant than when it's benign while being highly uncorrelated. For this condition to be fulfilled, the variables we chose are: *area_mean*, *concavity_mean* and *texture_mean* and we marked them as *chosen*,
3. For the last subset we also looked at the table 32 and picked the four variables as predictors regardless of their correlation. These are: *area_mean*, *concavity_mean*, *compactness_mean* and *concave.points_mean*, and described as *picked*.

3.2.1 Linear Discriminant Analysis

The first method that we will use is the LDA. Its purpose is to find a linear combination of features that characterizes or separates two or more classes of objects, which are the types of diagnosis. We used the method to fit the model on the training data and predict it's outcome on the test data. For each of the subsets, the confusion matrices look as follows:

Confusion matrix for LDA on the all subset

	B	M
predicted B	97	11
predicted M	1	62

Figure 37: Confusion matrix

The misclassification error for *all* subset is: $me \approx 0.07017$.

Confusion matrix for LDA on the chosen subset

	B	M
predicted B	98	16
predicted M	0	57

Figure 38: Confusion matrix

The misclassification error for *chosen* subset is: $me \approx 0.09356$.

Confusion matrix for LDA on the picked subset

	B	M
predicted B	97	17
predicted M	1	56

Figure 39: Confusion matrix

The misclassification error for *picked* subset is: $me \approx 0.10526$.

We can notice that the smallest misclassification error as well as the lowest false negative error is for the *all* subset of variables. Unfortunately the proportion of false negative to the false positive is high, what makes a relatively large number of the worst case scenario cases.

What is also interesting we can also present the decision boundaries of the LDA model between the variables that seem to be the most meaningful in our model: *area* and *concavity*:

As in the figure 40 the border is linear, the classification might not be as good as it would be if it was described by polynomial factors.

Decision boundary for LDA with area and concavity variables

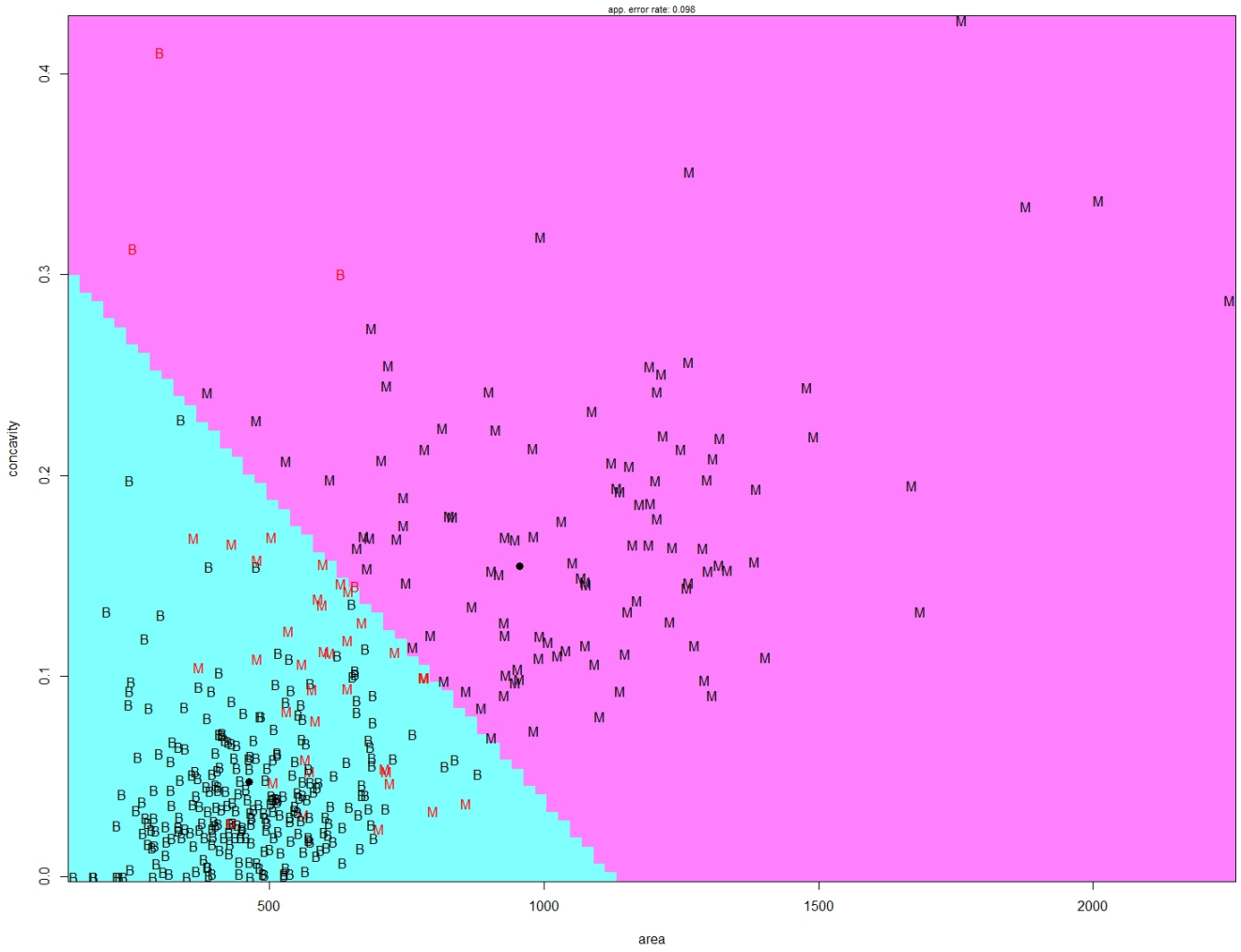

Figure 40: Figure of the decision boundary

3.2.2 Quadratic Discriminant Analysis

As the second method for classifying our data we will use the QDA, which is the more general version of LDA, as it allows the combination of features to be quadratic as well as linear. We proceeded in the similar way as previously, fitting the model, predicting the response and calculating the confusion matrices and misclassification errors.

Confusion matrix for QDA on the all subset

	B	M
predicted B	95	7
predicted M	3	66

Figure 41: Confusion matrix

The misclassification error for *all* subset is: $me \approx 0.05847$.

Confusion matrix for QDA on the chosen subset

	B	M
predicted B	97	12
predicted M	1	61

Figure 42: Confusion matrix

The misclassification error for *chosen* subset is: $me \approx 0.07602$.

Confusion matrix for QDA on the picked subset

	B	M
predicted B	92	10
predicted M	6	63

Figure 43: Confusion matrix

The misclassification error for *picked* subset is: $me \approx 0.09356$.

The pattern of values is much alike like in the LDA model, as the smallest error is for the *all* subset, and the biggest for *picked*. This time we limited a bit the number of false negative errors, and the fit of the data is more accurate than previously, what makes this method better.

As a confirmation we are going to show the decision boundaries figure for QDA model, for the same variables which were used previously – *area* and *concavity*.

Decision boundary for QDA with area and concavity variables

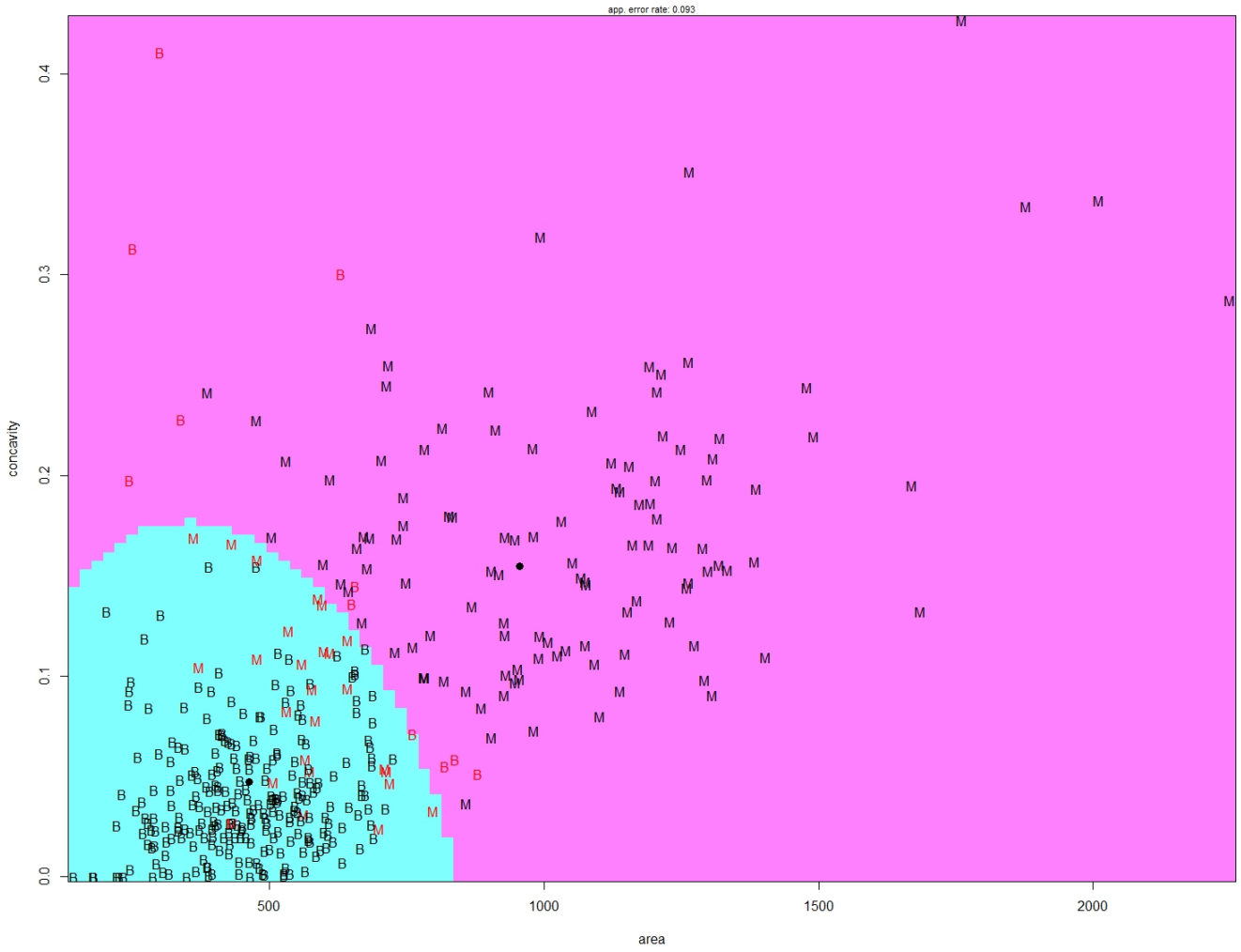

Figure 44: Figure of the decision boundary

The figure 44 shows that if the factors describing the predictors are of a higher order, the fit of the model to the data is just better.

3.2.3 k-Nearest Neighbours algorithm

Our next method of classyfing the data will be the k-NN algorithm, which is a non-parametric classification method. In this case we carry on in the same fashion and predict the outcome by the trained and fitted model. For the purpose of this task we picked the parameter k as equal to 5. The confusion matrices for the three subsets of variables look as follows:

Confusion matrix for k-NN on the all subset

	B	M
predicted B	93	16
predicted M	5	57

Figure 45: Confusion matrix

The misclassification error for *all* subset is: $me \approx 0.12281$.

Confusion matrix for k-NN on the chosen subset

	B	M
predicted B	93	15
predicted M	5	58

Figure 46: Confusion matrix

The misclassification error for *chosen* subset is: $me \approx 0.11695$.

Confusion matrix for k-NN on the picked subset

	B	M
predicted B	88	14
predicted M	10	59

Figure 47: Confusion matrix

The misclassification error for *picked* subset is: $me \approx 0.14035$.

This time the situation is a little bit different. The misclassification error takes the smallest value for the *chosen* subset, indicating that a model conctructed in this way is th most accurate. On the other hand, the number of false negative errors is in the lowest in the last model. However this difference is insignificant, thus we can assume that the middle model provides the best results, despite the k-NN method overally being the worst in terms of accuracy among the methods presented earlier.

We were also wondering whether the accuracy would improve if we took a different value of k for our model, so for the *all* subset model we provided the k from the range [1,20] and checked the results.

Table with the missclasification error for different values of k

	k_s	errors
1	1	0.111111111111111
2	2	0.128654970760234
3	3	0.128654970760234
4	4	0.111111111111111
5	5	0.111111111111111
6	6	0.12280701754386
7	7	0.116959064327485
8	8	0.116959064327485
9	9	0.116959064327485
10	10	0.12280701754386
11	11	0.12280701754386
12	12	0.128654970760234
13	13	0.116959064327485
14	14	0.12280701754386
15	15	0.116959064327485
16	16	0.12280701754386
17	17	0.116959064327485
18	18	0.12280701754386
19	19	0.116959064327485
20	20	0.116959064327485

Figure 48: Table of errors

As we observe on figure 48, the differences in the errors' values isn't substantial, but lesser values appear while we consider the smaller values of k. Thus, picking initially $k = 5$ is justified.

3.2.4 Logistic Regression

Another method that we will utilize is Logistic Regression. In order for it to work properly, we have to change the categorical responses in out data into binary ones, meaning that for this model the benign category would be set as 0, and the malignant as 1. The next steps of classification are the same as previously while using the same variable subsets. Thus, the confusion matrices and erorrs are:

Confusion matrix for Logistic Regression on the all subset

	0	1
predicted B	98	0
predicted M	0	73

Figure 49: Confusion matrix

The misclassification error for *all* subset is: $me = 0$!

Confusion matrix for Logistic Regression on the chosen subset

	0	1
predicted B	96	9
predicted M	2	64

Figure 50: Confusion matrix

The misclassification error for *chosen* subset is: $me \approx 0.06432$.

Confusion matrix for Logistic Regression on the picked subset

	0	1
predicted B	89	9
predicted M	9	64

Figure 51: Confusion matrix

The misclassification error for *picked* subset is: $me \approx 0.10526$.

This time we notice that for the *all* subset there is no misclassification error, what implies that the accuracy is perfect. All the responses are corretly matched. For the remaining subsets the errors are quite higher, but still the number of false negative errors isn't as high as in the previous methods, making the Logistic Regression one of the most viable tool for this assignment.

3.2.5 ROC curves and AUC values

As we already described four methods, we can prepare the ROC curves for these methods and calculate AUC for them. For each subset of variables we created a graph that shows the behaviour of these curves.

ROC curves for different methods of the all subset

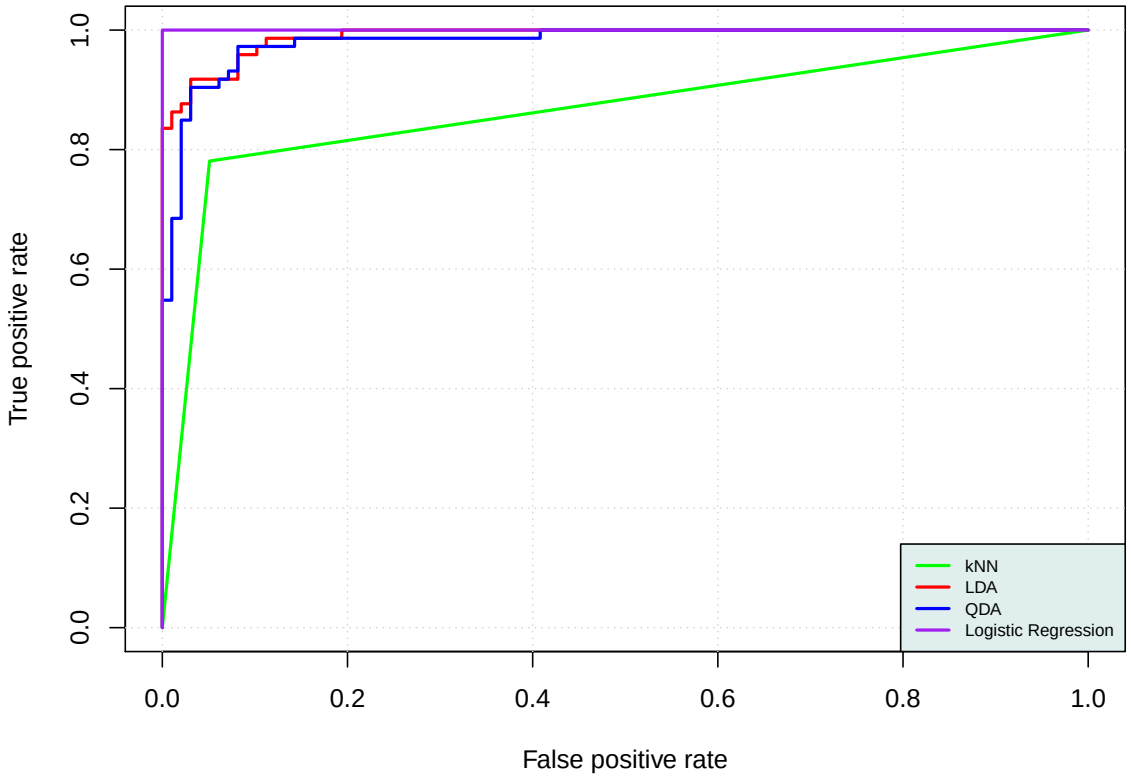

Figure 52: Figure of the ROC curves

The AUC values are respectively:

- 1 for Logistic Regression
- 0.9892 for LDA
- 0.9808 for QDA
- 0.8649 for kNN

What catches an eye on the figure 52 is the curve of kNN algorithm, it implies that this method doesn't work particularly well for the analyzed data. The other methods are better for this task as their AUC values are close to 1.

ROC curves for different methods of the chosen subset

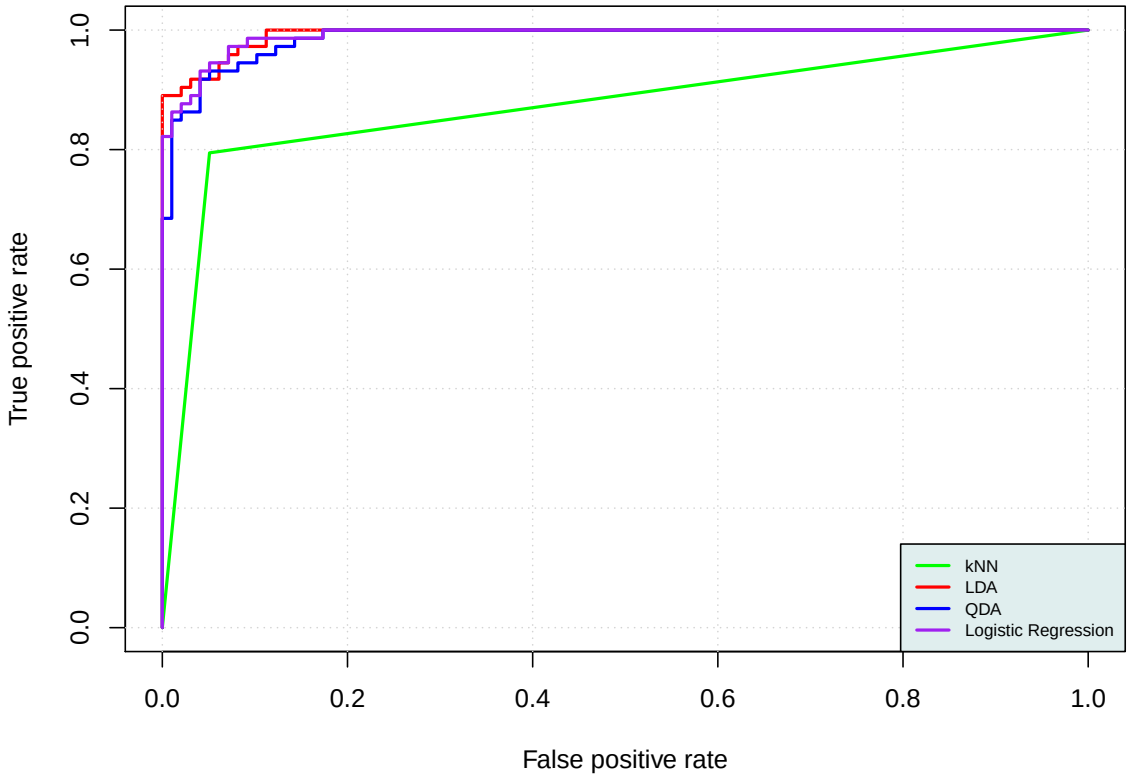

Figure 53: Figure of the ROC curves

The AUC values are as follows:

- 0.9909 for Logistic Regression
- 0.9925 for LDA
- 0.9866 for QDA
- 0.8717 for kNN

For the figure 53 the pattern is the same, as the AUC value for kNN algorithm stands out as a significantly smaller than others, while the Logistic Regression having the best fit. Besides the LR, ROC curves take the similar shape as in the previous graph.

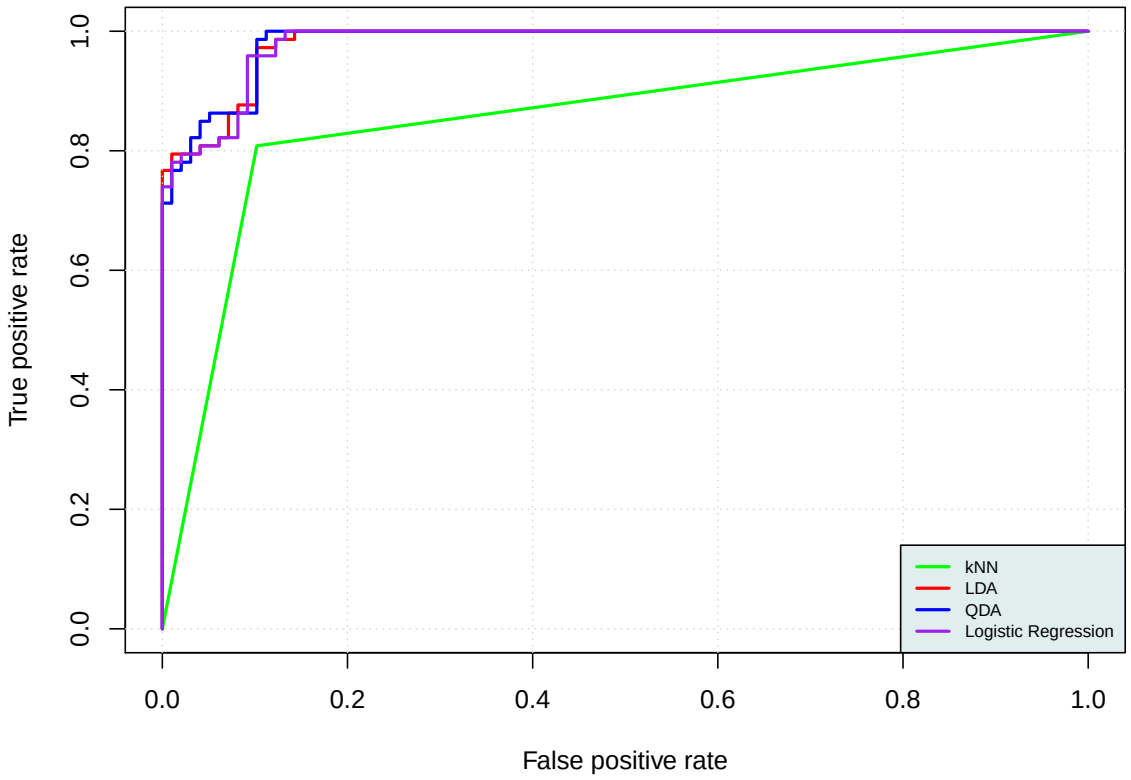

Figure 54: Figure of the ROC curves

The AUC values:

- 0.9806 for Logistic Regression
- 0.9808 for LDA
- 0.9820 for QDA
- 0.8530 for kNN

For the *picked* subset the figure 54 shows that the best fit seems to be assigned to QDA, as its AUC value is slightly higher than for LDA and LR. The ROC curves for all the methods look almost the same as for the preceding examples.

3.2.6 Classification Trees

Another method that will be useful for data classification are Classification Trees which are structural mappings of binary decisions that lead to a decision about the class. For this method the steps are a bit diverse. Firstly, we create the tree basing on the training data. Then, we need to find the optimal *cp* parameter in order to prune the tree, what optimizes the process of classification. When it's complete we can predict the response using the test data and compare it with the true outcome. Similarly as before, we'll be using the three subsets of variables. At first, we can compare the shapes of original trees and the trees after pruning:

Comparison of trees of all subset

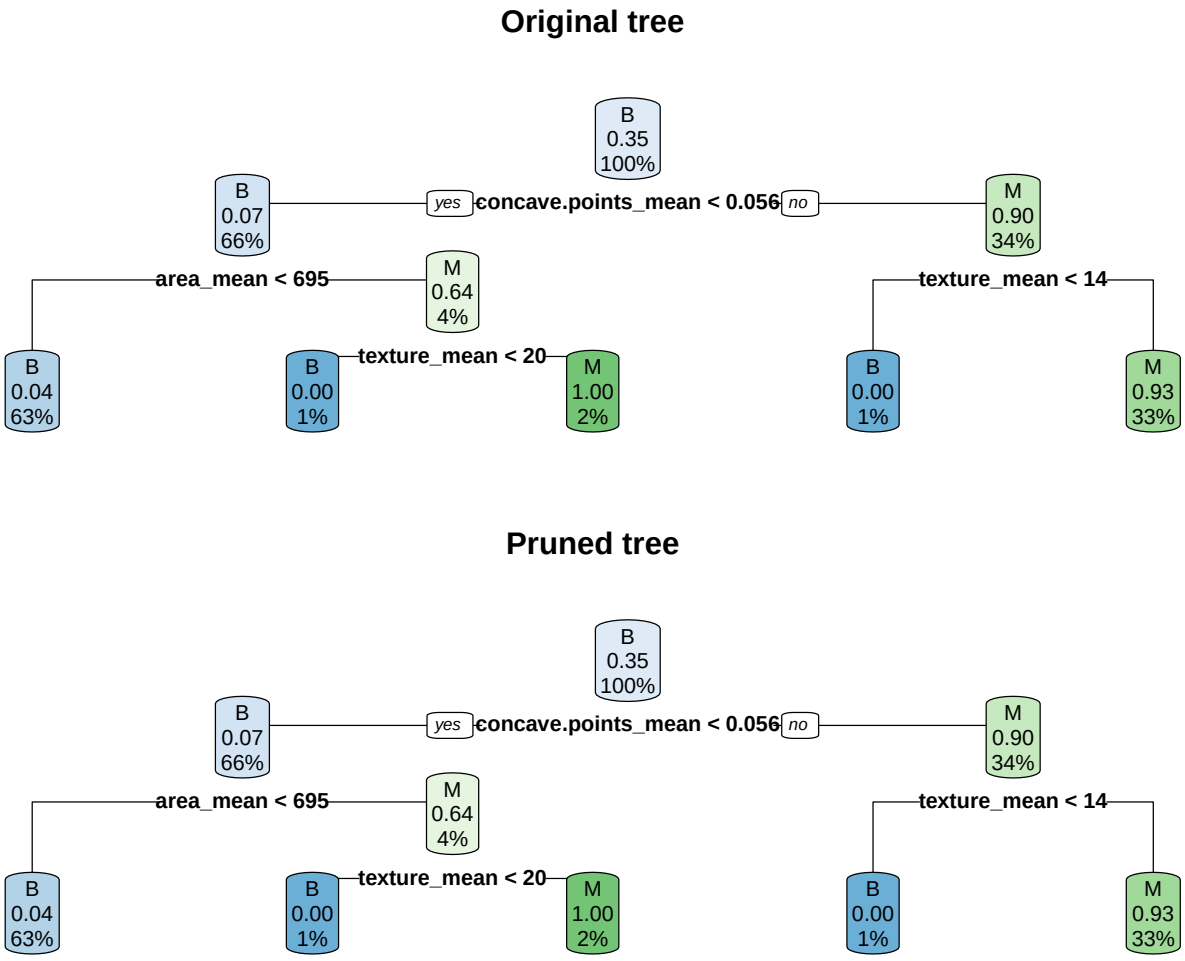

Figure 55: Classification trees

Comparison of trees of chosen subset

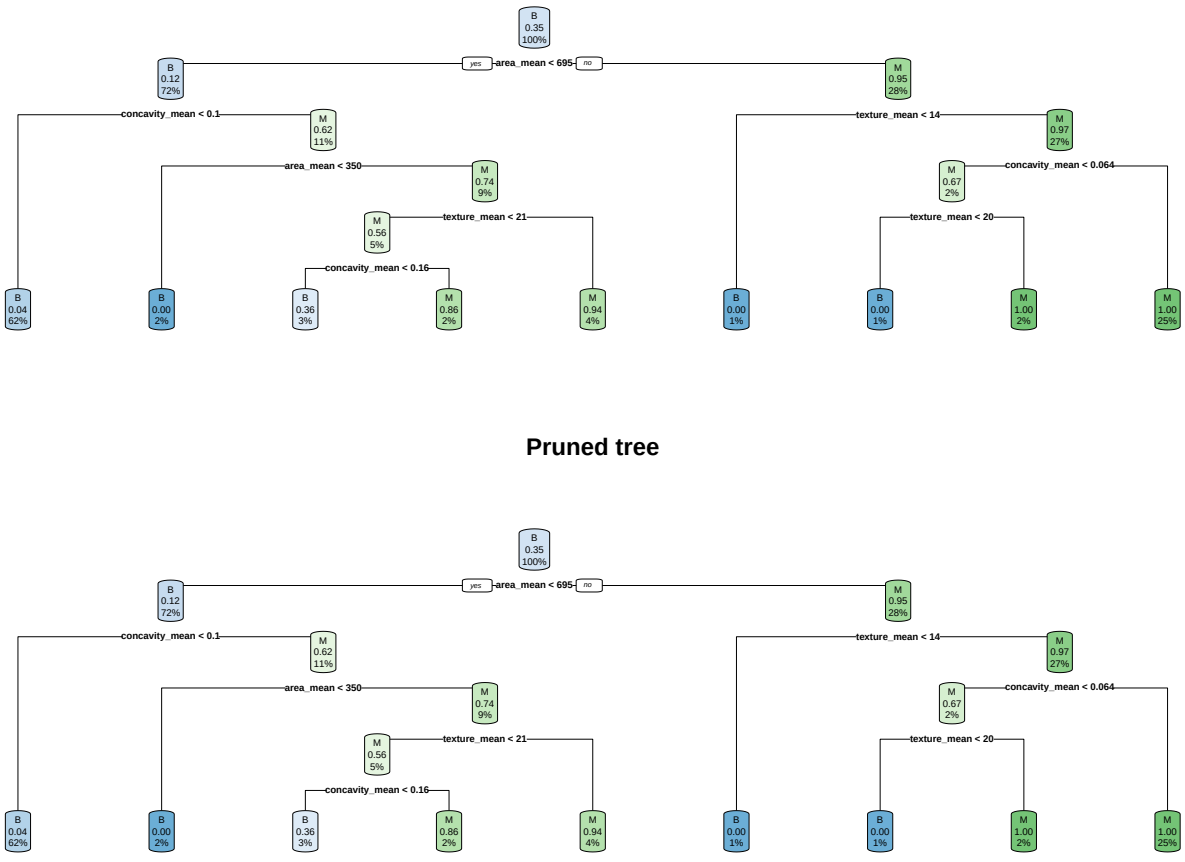

Figure 56: Classification trees

On figures 56 and 55 we can notice that the pruning didn't change the shape of the tree for both *all* and *chosen* subsets, so in these cases we can omit the process of pruning and predict the response only for the original tree.

Comparison of trees of picked subset

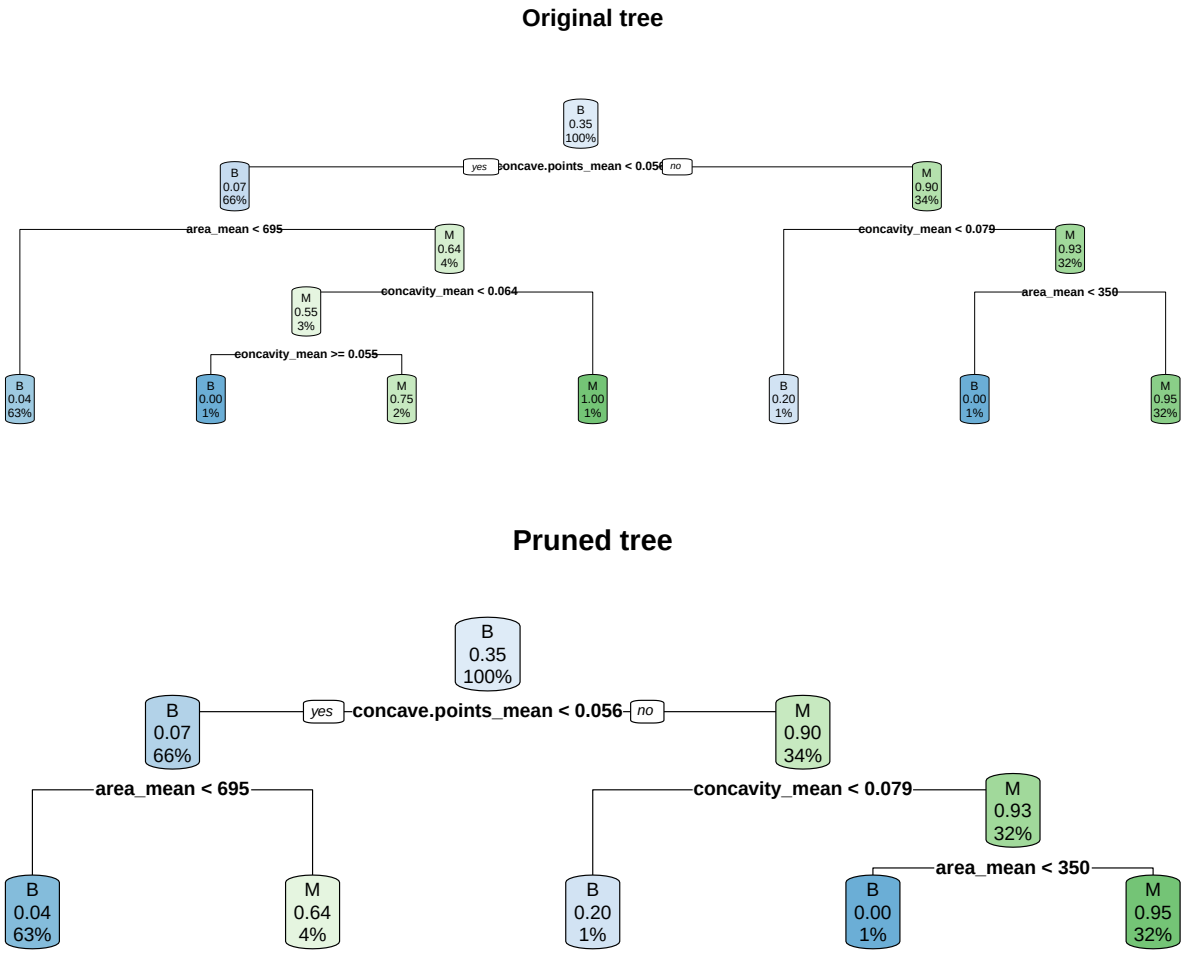

Figure 57: Classification trees

For the *picked* subset, the figure 57 the process of pruning resulted in the reduction of the tree's size. In this case we can conduct the preduiction process for both – the original and pruned tree.

For classification trees the confusion matrices look respectievly.

Confusion matrix for Classification Tree on the all subset

	B	M
predicted B	96	11
predicted M	2	62

Figure 58: Confusion matrix

The misclassification error for *all* subset is: $me \approx 0.076$.

Confusion matrix for Classification Tree on the chosen subset

	B	M
predicted B	96	11
predicted M	2	62

Figure 59: Confusion matrix

The misclassification error for *chosen* subset is: $me \approx 0.076$.

Confusion matrix for Classification Tree on the picked subset

	B	M
predicted B	91	7
predicted M	7	66

Figure 60: Confusion matrix

The misclassification error for *picked* subset is: $me \approx 0.08187$.

Confusion matrix for pruned Classification Tree on the picked subset

	B	M
predicted B	90	6
predicted M	8	67

Figure 61: Confusion matrix

The misclassification error for pruned *picked* subset is: $me \approx 0.08187$.

The accuracy of classification for the first two subsets seem to be higher than for the *picked* subset in terms of misclassification error, but the number of false negative errors for the last two trees is slightly lower. If we consider the original and pruned versions of trees for *picked* subset, we can claim that the pruned one performs better in terms of minimising the FP error, although the number of correctly classified cases is equal. With such small differences it's difficult to pick one of these models and consider it as the best among others.

3.2.7 Support Vector Machines

The following method can efficiently perform a non-linear classification, but we can use it for the classification of our data. This time we'd rather compare the performance of the different types of SVMs than compare one of them among the subsets of variables. For that purpose for the *all* subset we created three models – linear (with cost parameter equal to 1), polynomial (with degree 2) and radial (with gamma parameter as 0.1). For each of the models the steps are the same as for the previously mentioned methods. We predict the response on the test data and obtain confusion matrices, while the response also had to be converted to a binary one with malignant denoted as 1 and benign as 0.

Confusion matrix for linear SVM on the all subset

	0	1
predicted B	98	0
predicted M	0	73

Figure 62: Confusion matrix

The misclassification error for linear SVM is: $me = 0$.

Confusion matrix for polynomial SVM on the all subset

	0	1
predicted B	98	3
predicted M	0	70

Figure 63: Confusion matrix

The misclassification error for polynomial SVM is: $me \approx 0.01754$.

Confusion matrix for radial SVM on the all subset

	0	1
predicted B	98	0
predicted M	0	73

Figure 64: Confusion matrix

The misclassification error for radial SVM is: $me = 0$.

Taking the confusion matrices and misclassification errors into consideration, the SVMs are highly accurate methods in classification of the data. Only the three observations for the polynomial model were wrongly classified and for the remaining both we got a perfect match with the true response.

3.2.8 Bagging

The last method that we'll be using in the classification is Bagging. In this case we return to the previous approach where we analyzed the method on three different subsets of variables. As described boefore, we fit the models on training data and predict the response using the test part. That leads to the follwoing confusion matrices:

Confusion matrix for Bagging on the all subset

	0	1
predicted B	98	0
predicted M	0	73

Figure 65: Confusion matrix

The misclassification error for Bagging for *all* subset is: $me = 0$.

Confusion matrix for Bagging on the chosen subset

	0	1
predicted B	93	6
predicted M	5	67

Figure 66: Confusion matrix

The misclassification error for Bagging for *chosen* subset is: $me \approx 0.06432$.

Confusion matrix for Bagging on the picked subset

	0	1
predicted B	93	6
predicted M	5	67

Figure 67: Confusion matrix

The misclassification error for Bagging for *picked* subset is: $me \approx 0.06432$.

We can consider this classifier as also quite accurate, as for the *all* subset the match was perfect and the misclassification error for the other ones was really low. The number of false negative errors is also low what makes bagging a good tool for performing such classification.

4 Discussion of the results and conclusions

The exploratory approach to the data allowed us to check the distribution of each of the features, visualise it and helped to understand better the essence of the analyzed problem. The division of the data into two categories showed, that in general if the tumor is malignant, majority of the factors that described it are higher than in the case of tumor being benign. The analysis of this dependence and finding out the correlation between each parameters enabled us to select the subsets of features that we thought had an effect on the diagnosis. In the classification section we have presented the results of our analysis. The set of data that we used wasn't too extensive, but we managed to thoroughly describe the variables and values and perform different classification assignments for the various data subsets. It seems that the correct recognition of the type of the cancer can have a big affect on the patient's health and life, so we put a great effort into determining the best method to identify whether the tumor is malignant or benign. To sum up and have everything in one place, we created a table with all the misclassification errors for each subset of variables.

Summary of the errors for all subset

	misclassification error
error.lda.all	0.0701754385964912
error.qda.all	0.0584795321637427
error.knn.all	0.12280701754386
error.log.all	0
error.tree.all	0.0760233918128655
error.svm.linear.all	0
error.svm.polynomial.all	0.0175438596491228
error.svm.radial.all	0
error.bagging.all	0

Figure 68: Table with the summary of errors

As we notice, there are as many as four methods that perfectly predicted the category of the tumor for this particular subset of features. That makes it really difficult to choose one of them to be the best one, but we can also take a look at the error statistics for the two other subsets.

Summary of the errors for chosen subset

	misclassification error
error.lda.chosen	0.0935672514619883
error.qda.chosen	0.0760233918128655
error.knn.chosen	0.116959064327485
error.log.all	0.064327485380117
error.tree.chosen	0.0760233918128655
error.bagging.chosen	0.064327485380117

Figure 69: Table with the summary of errors

Although we didn't use SVMs for this subset of features, we can spot that the error is smallest for Logistic Regression and Bagging. If we also check the value of false negative error which is accordingly 9 and 6, we can claim that in this case the Bagging method performs best. The sumary for the last subset is as follows:

Summary of the errors for picked subset

	misclassification error
error.lda.picked	0.105263157894737
error.qda.picked	0.0935672514619883
error.knn.picked	0.140350877192982
error.log.picked	0.105263157894737
error.tree.picked	0.0818713450292398
error.tree.pruned.picked	0.0818713450292398
error.bagging.picked	0.064327485380117

Figure 70: Table with the summary of errors

In this case the best performing method is also Bagging having the least misclassification error of all the methods included in the table. The number of false negative error is also the same as previously and equal to 6.

All in all, we are able to state that the best methods for the classification of the used medical data are SVMs – in patricular the linear and radial, and Bagging. The application of these approaches resulted in the best performance among other techniques and provided the most acurate outcomes.