Data Mining

Course project

Marta Kawalko (229955), Zuzanna Materny (229932)

Application of data mining techniques on *Breast Cancer Wisconsin* data set

Contents

1 Introduction

The project goal is to use data mining methods to perform complete analysis of selected data. At the same time we will familiarize with the real problem thoroughly.

Breast cancer is one of the most common cancers women are facing. More than every tenth woman have suffered from the disease. Many types of breast cancer are not difficult to diagnose, but we have to be constantly under the guidance of a doctor. A breast self-examination (BSE) is one of the most important techniques that need to be fulfilled by each woman, at least once in two months time starting at the age of 18. Such manner may stop cancer from spreading and allows to recognize the first signs of the disease.

In the project we will analyze breast cancer data obtained from the University of Wisconsin Hospitals ([1]). The data was collected in the years 1989-1991 and contains the following information:

	Attribute	Domain
1.	Sample code number	id number
2.	Clump Thickness	1 - 10
3.	Uniformity of Cell Size	1 - 10
4.	Uniformity of Cell Shape	1 - 10
5.	Marginal Adhesion	1 - 10
6.	Single Epithelial Cell Size	1 - 10
7.	Bare Nuclei	1 - 10
8.	Bland Chromatin	1 - 10
9.	Normal Nucleoli	1 - 10
10.	Mitoses	1 - 10
11.	Class:	(2 for benign, 4 for malignant)

2 Preliminary data processing

Before performing analysis we need to prepare and clean the data so the way it is stored in our data frame is both correct and convenient for analyzing.

2.1 Reading and understanding the data

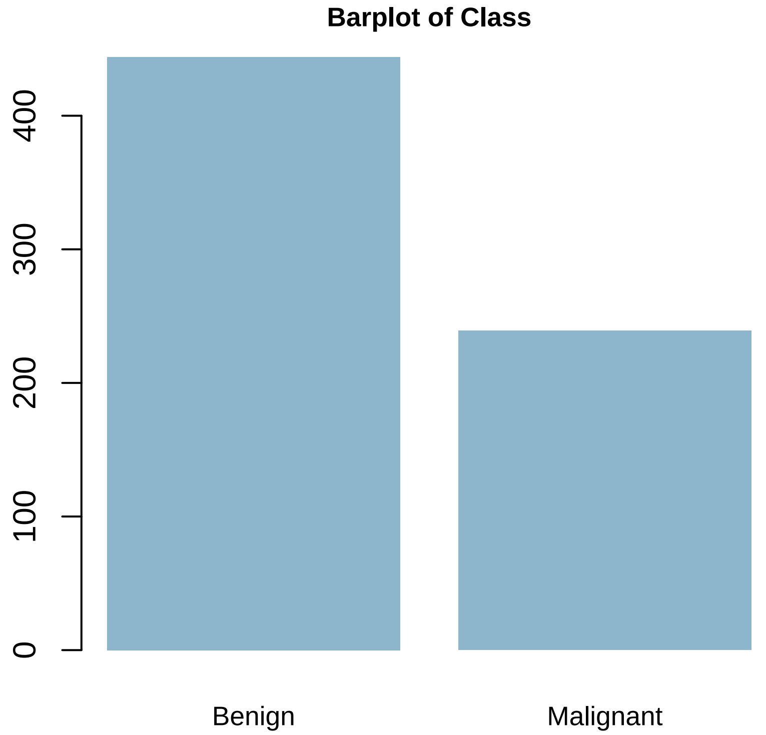
To make sure we analyze the data properly, we have to make sure to understand all the information included in the dataset. At first sight it seems that most of the variables are categorical qualitative and ordinal. The values of variables are natural numbers from 1 to 10 and the class of cancer is showing values from the set $\{2, 4\}$, which tells us if the cancer is benign or malignant. The smaller the values of those variables are, the closer the cancer is to being benign. Only the first variable in the set − code number − is the unique identificator of diagnosed woman. This feature is not useful in out analysis, because it is the identifier-type feature and we are not going to check the objective women one by one or analyse their conditions individually. In the dataset there is no other reduntant feature and all the values are from the fixed set. As mentioned before the variables are cathegorical qualitative and ordinal, we should make sure they are coded as factors in R. But this approach limits our possibilities and ways to analyze the dataset so we will treat the variables also as numeric being very careful at the same time.

2.2 Handling missing values

We have noticed 16 missing values of the variable `Bare.Nuclei` coded as "?". It stands for $2,3\%$ of all entries. They are situated randomly among other records and the remaining attributes have diverse values, thus there is no need to inspect them with special attention and we decided to just remove those cases. Our final dataset contains 683 observations of 10 variables.

2.3 Identification and interpretation of outliers

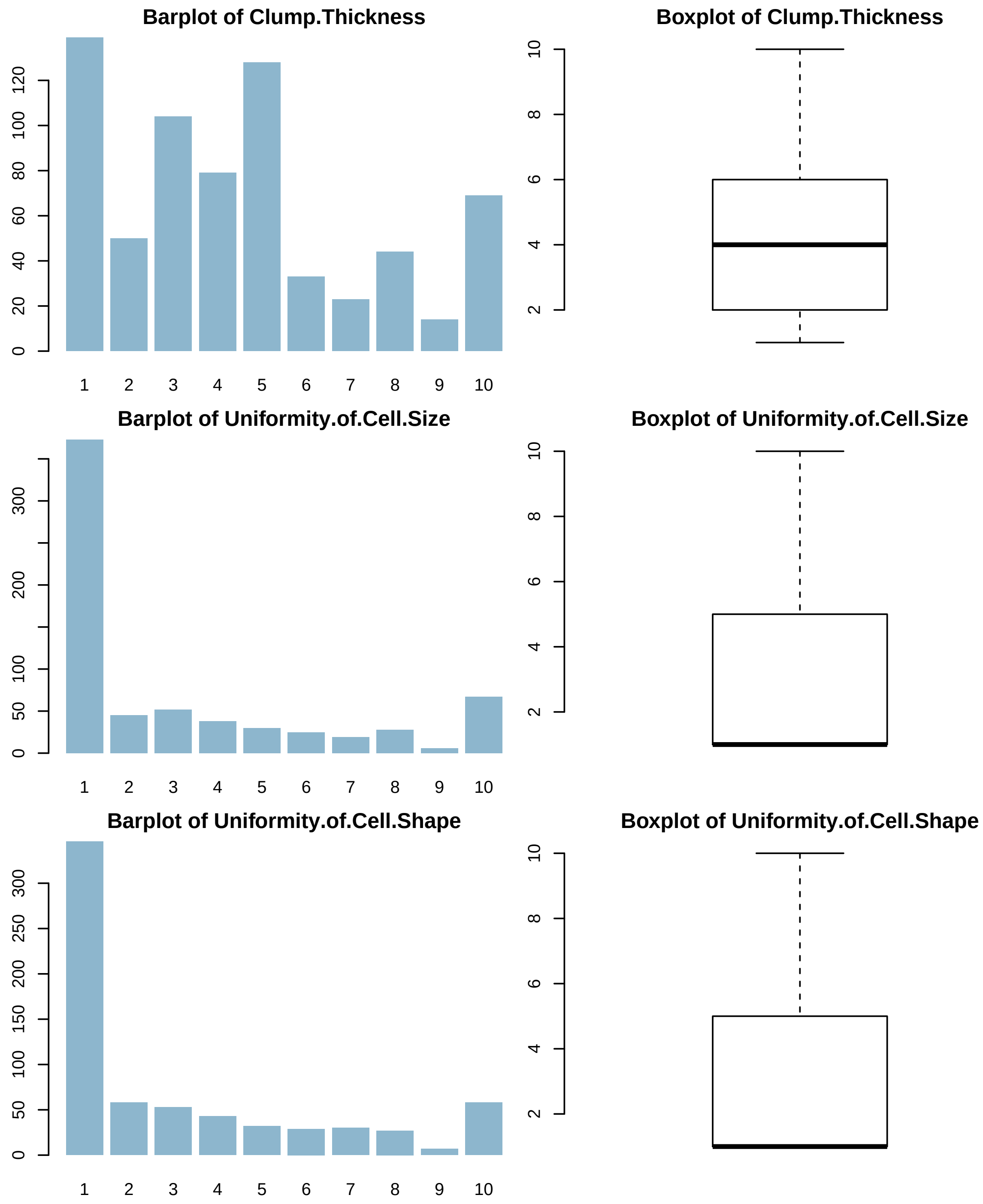
As all the values are natural numbers from 1 to 10 and there are no inconsistencies in the dataset.



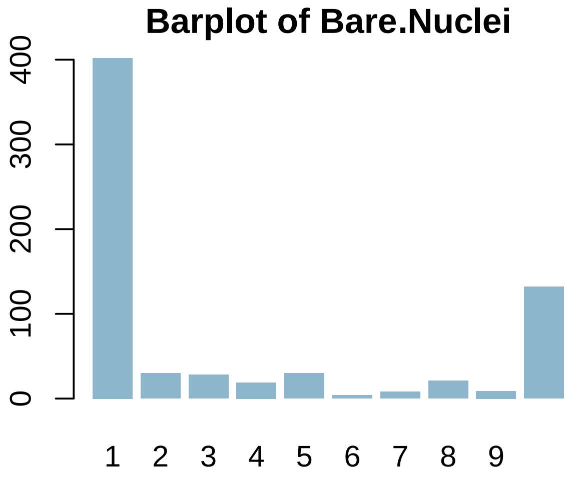
The first barplot tells us a lot about the malignancy of the cancer among the examined women. The benign cancer is more popular but 35% of women suffer from more dangerous cancer – malignant.

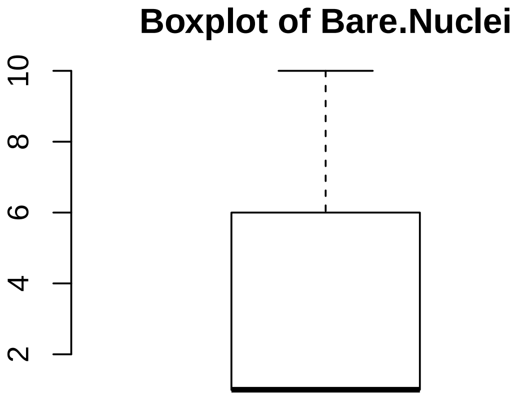
One of the first signs of breast cancer a woman can feel herself is the clump itself. The thicker it is, the easier it is to feel, so the clump thickness is very important information.

The values of thickness are quite varied. The boxplot below does not reveal any outliers.

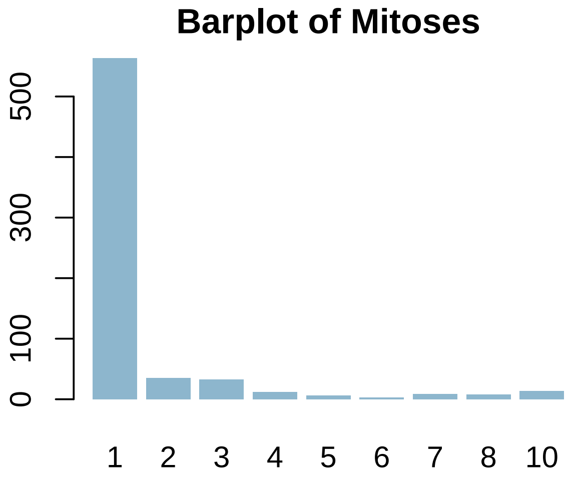


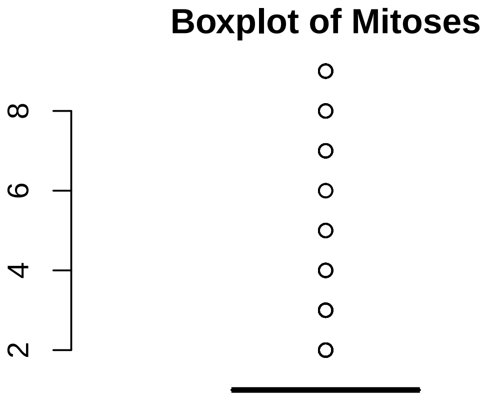
Let's have a look at the uniformity of cell shape and size. Those two variables have similar barplots and deepening of one of those two features seems to be correlated with with the upgrowth of the second one, but we will check the correlation in the next section. The median of the values is much smaller than the mean, because the most of the uniformity cell shapes and sized is classified as 1. There are no outliers for those two variables.



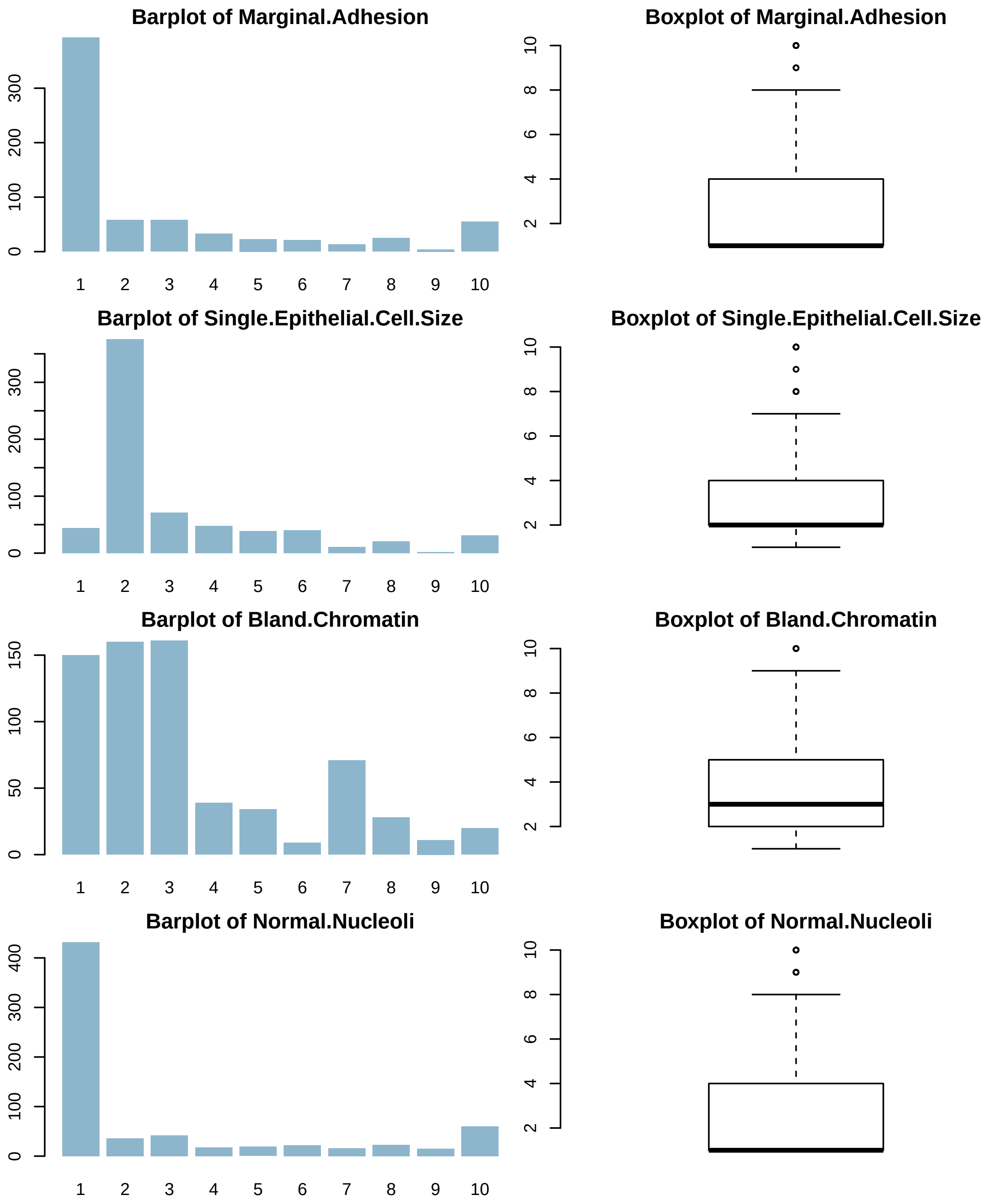


The bare nuclei variable distribution looks similar to the feature describing the uniformity of the cells. The correlation between those will be discusses in the following sections.





Mitoses is the most critical variable in case of outliers. This is due to the large number of values 1 along with just a few values of the other qualities of mitoses - a special type of cell division that results in two daughter cells each having the same kind and number of chromosomes as the parent nucleus. It means that the cell division at the high level is not very popular among the examined women.



All the plots above present distribution of values for each variable. Most of them are concentrated around 1 and each of them is on average smaller than 5. For many features boxplots indicate outstanding values. In this medical case we cannot infer that they are errors. They probably correspond to some symptoms of ilness. If we check the people who got at least one score of 10, we will see that 204 of them (97,6%) have a malignant tumor. Just to remind, in the whole data frame we have 239 malignant cases.

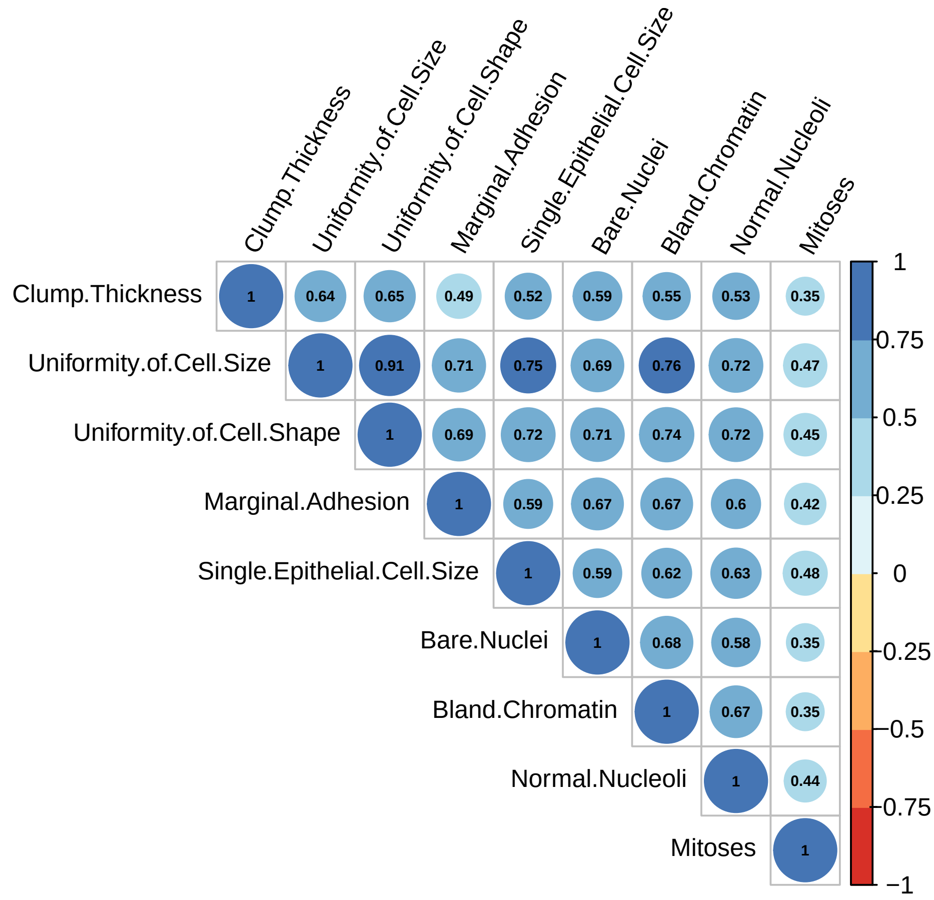
```
table(df[apply(df, 1, function(row) any(row == 10)), 'Class'])

##
##    Benign Malignant
##         5       204
```

Thus, in the further analysis we should keep in mind that big, outstanding values are crucial for the classification.

3 Exploratory data analysis

In order to perform classification more efficiently it is necessary to know as much as possible about the features' properties. The basic information and the distribution of each variable separately was presented in the previous section. Now we will focus on the relationships between the features.

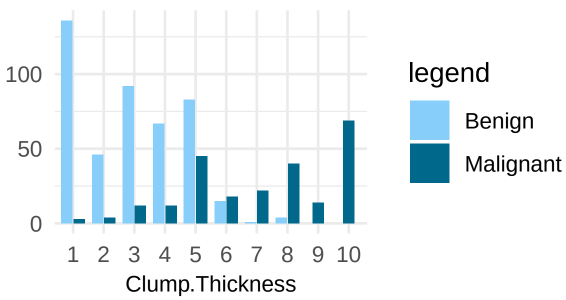
One of the things we can check is a correlation between all the variables describing the stage of the cancer signs — not the cancer itself, so all the variables except `Class`.

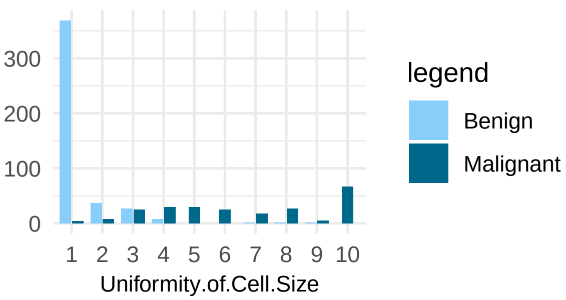


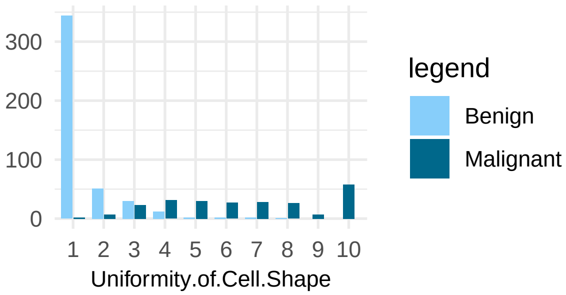
If two variables are highly correlated, we can omit one of them while building, for example, linear regression model. However, since many of the variables take the value 1 the most often, the correlation does not provide strongly relevant information. We will not rely on it in the further classification analysis, but we will check whether it improves our models.

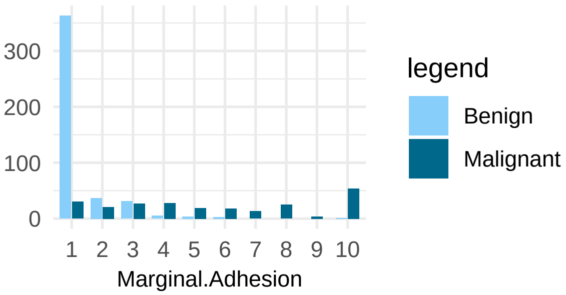
The biggest correlation can be noticed between `Uniformity.of.Cell.Shape` and `Uniformity.of.Cell.Size` and is equal to 0.91. Variable `Mitoses` has the lowest correlation with the others.

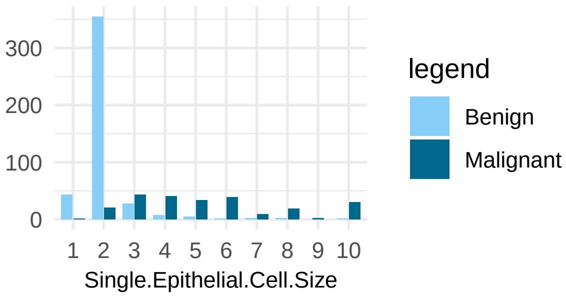
Also, it is good to know whether any of the attributes has a discriminating ability. We inspect that visually using barplots.

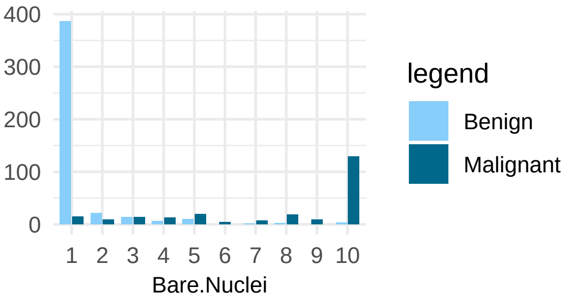

Barplot of Clump.Thickness

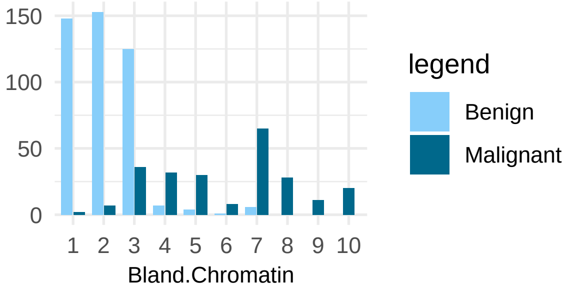

Barplot of Uniformity.of.Cell.Size

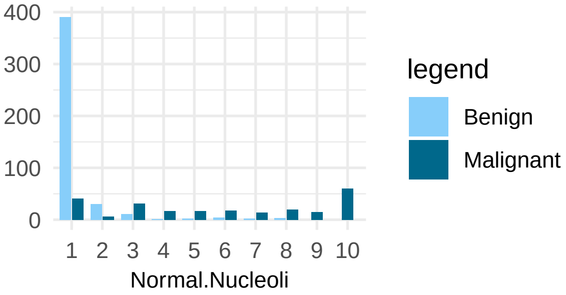

Barplot of Uniformity.of.Cell.Shape

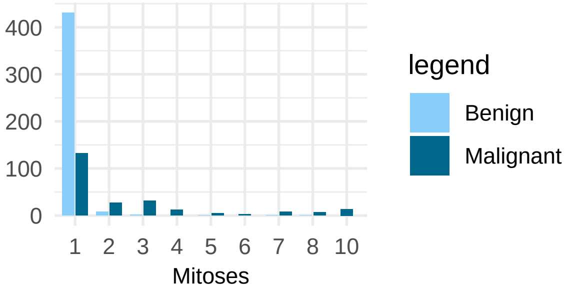

Barplot of Marginal.Adhesion


Barplot of Single.Epithelial.Cell.Size


Barplot of Bare.Nuclei


Barplot of Bland.Chromatin


Barplot of Normal.Nucleoli


Barplot of Mitoses

Clear division is not visible in any of the variables, although in general small values stand for benign cancer, while extremely big indicate almost always malignant one. It is also supported by measures of central tendency — mean and median.

Class	Clump.Thickness	Uniformity.of.Cell.Size	Uniformity.of.Cell.Shape	Marginal.Adhesion
Benign	2.96	1.31	1.41	1.35
Malignant	7.19	6.58	6.56	5.59

Class	Single.Epithelial.Cell.Size	Bare.Nuclei	Bland.Chromatin	Normal.Nucleoli	Mitoses
Benign	2.11	1.35	2.08	1.26	1.07
Malignant	5.33	7.63	5.97	5.86	2.54

Table 1: **Mean of the variables with respect to Class**

Class	Clump.Thickness	Uniformity.of.Cell.Size	Uniformity.of.Cell.Shape	Marginal.Adhesion
Benign	3.00	1.00	1.00	1.00
Malignant	8.00	6.00	6.00	5.00

Class	Single.Epithelial.Cell.Size	Bare.Nuclei	Bland.Chromatin	Normal.Nucleoli	Mitoses
Benign	2.00	1.00	2.00	1.00	1.00
Malignant	5.00	10.00	7.00	6.00	1.00

Table 2: **Median of the variables with respect to Class**

Only `Mitoses` behaves differently. Most of the malignant cancers takes here value 1. But we should also remember that the cell division process does not necessarily mean that the cancer is progressing, mitoses at a high level can albo be noticed for a healthy person and at a low level can be observed for burdened with a tumor.

	Benign	Malignant
1	431	132
2	8	27
3	2	31
4	0	12
5	1	5
6	0	3
7	1	8
8	1	7
10	0	14

We remember that transformations may change important data characteristics. But as we look at out data set, we can see the set of values for all the features (except for the class) is the same and is not very divergent. We can only consider more advanced transformations like feature selection or feature aggregation, which we take care of in the next chapter. At the moment we can say that our set has a good data quality.

4 Classification

We will carry out the classification to check the models and predict our qualitative variables — classify the malignancy of tumor. In our case we have a binary classification task, because there are only two options: benign and malignant. Our goal is to construct a decision rule $G(\underline{x})$ which for any observation $\underline{x} \in \mathcal{X}$ will assign membership to one of the classes from the set $\mathcal{G}$. Based on the training set we will try to recognize the the relationship between the variables and use this knowledge to predict the further values.

4.1 Feature selection

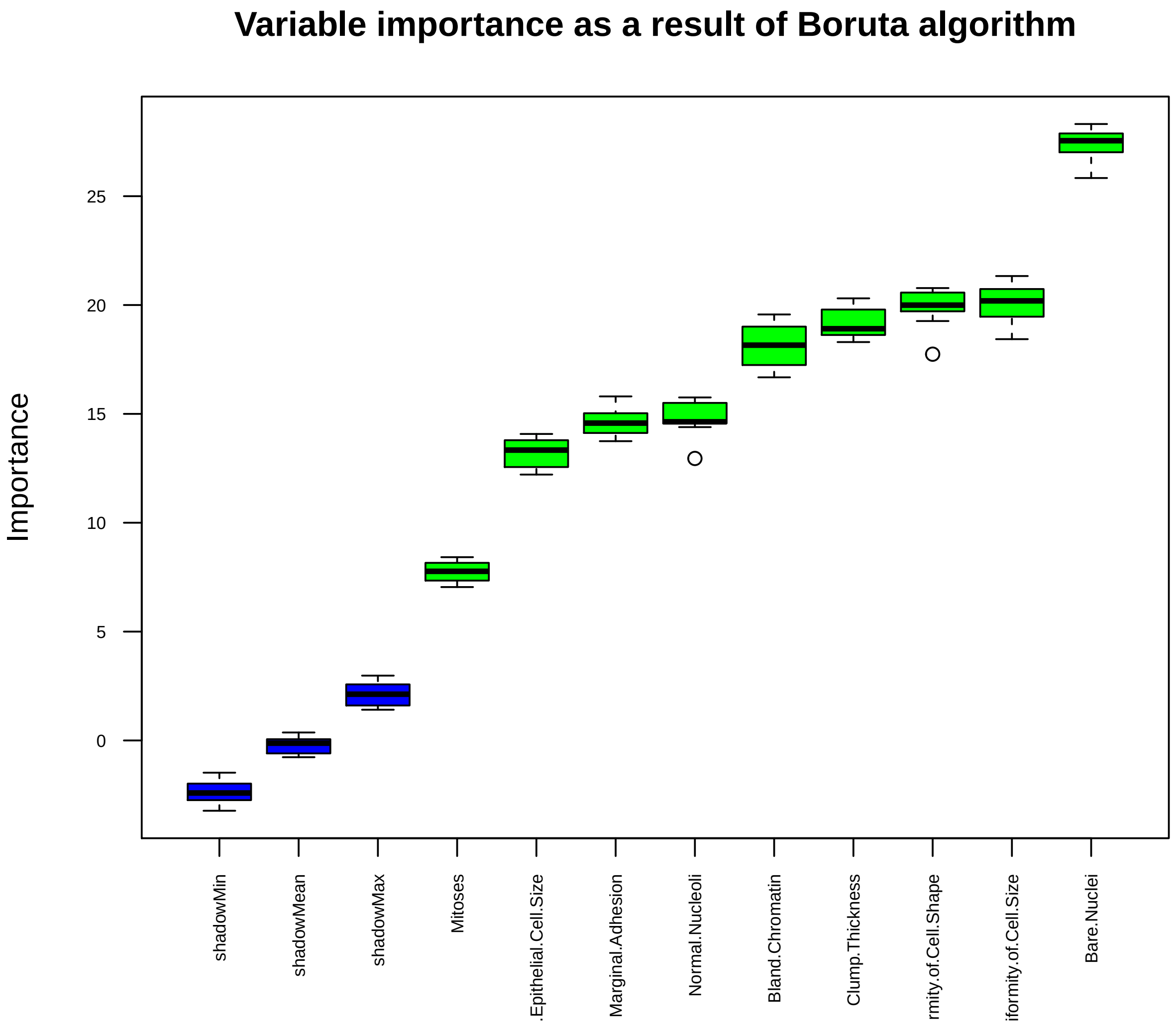
As we have learnt during the laboratiories, the selection of features can be achieved in two ways: One is to rank features according to some criterion and select the top k features, and the other is to select a minimum subset of features without learning performance deterioration. In other words, subset selection algorithms can automatically determine the number of selected features, while feature ranking algorithms need to rely on some given threshold to select features.

In our work we will try both ways to be sure that the chosen variables present the best data set to draw the correct conclusions and not to omit the important information.

We will split the data into the training and testing set. We will stick to this split throughout the whole feature selection analysis and then classification.

4.1.1 Boruta

The first method we apply to our data is Boruta, a wrapper method built around the random forest classification algorithm. It creates shadow attributes that help decide which of them are important. If the real attributes have the importance much lower than the shadow attributes, then they are considered irrelevant. As the result we will see if the variables are Important, Tentative or Rejected.

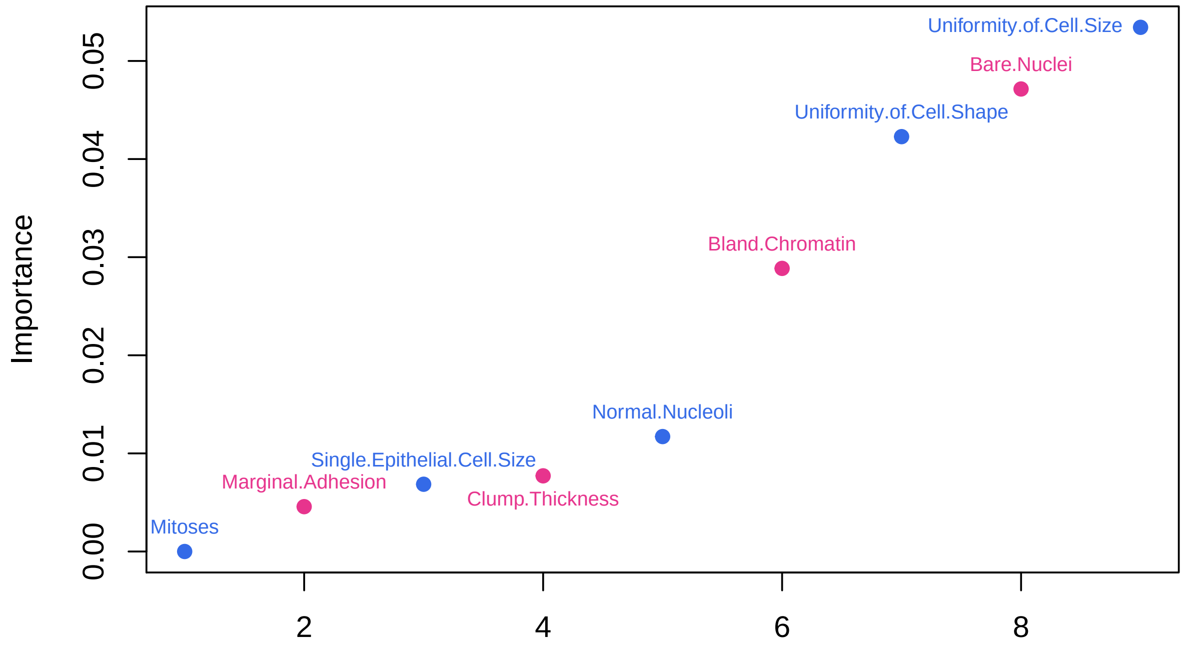


That's a very surprising result. The Boruta algorithm returned the information that all 9 variables are considered important! All the green boxplots corresponding to our variables are higher than blue boxplots (shadow attributes). Of course their importance differ and `Bare.Nuclei` is considered the most important, while `Mitoses` has the lowest value of importance. But still, all 9 features are said to be relevant.

```
getSelectedAttributes(boruta)

## [1] "Clump.Thickness"             "Uniformity.of.Cell.Size"
## [3] "Uniformity.of.Cell.Shape"    "Marginal.Adhesion"
## [5] "Single.Epithelial.Cell.Size" "Bare.Nuclei"
## [7] "Bland.Chromatin"             "Normal.Nucleoli"
## [9] "Mitoses"
```

4.1.2 Random Forest method with conditional inference trees

For this method we will use `cforest` function from `party` package based on recursive partitioning. As we have only 9 variables, we will take 3 as number of input variables randomly sampled as candidates at each node of the tree in forest. Our forest will contain 20 trees. Different colours were chosen for the clearness of the output.



As the result of the method we can see that `Uniformity.Of.Cell.Size` characterizes the biggest importance. For Boruta algirthm this variable has the second highest value of importance so we can be sure that this is defnitely a relevant feature. On the other hand, `Mitoses`, which was also at the bottom of Boruta importance hierarchy, is distinguished by the low relative importance.

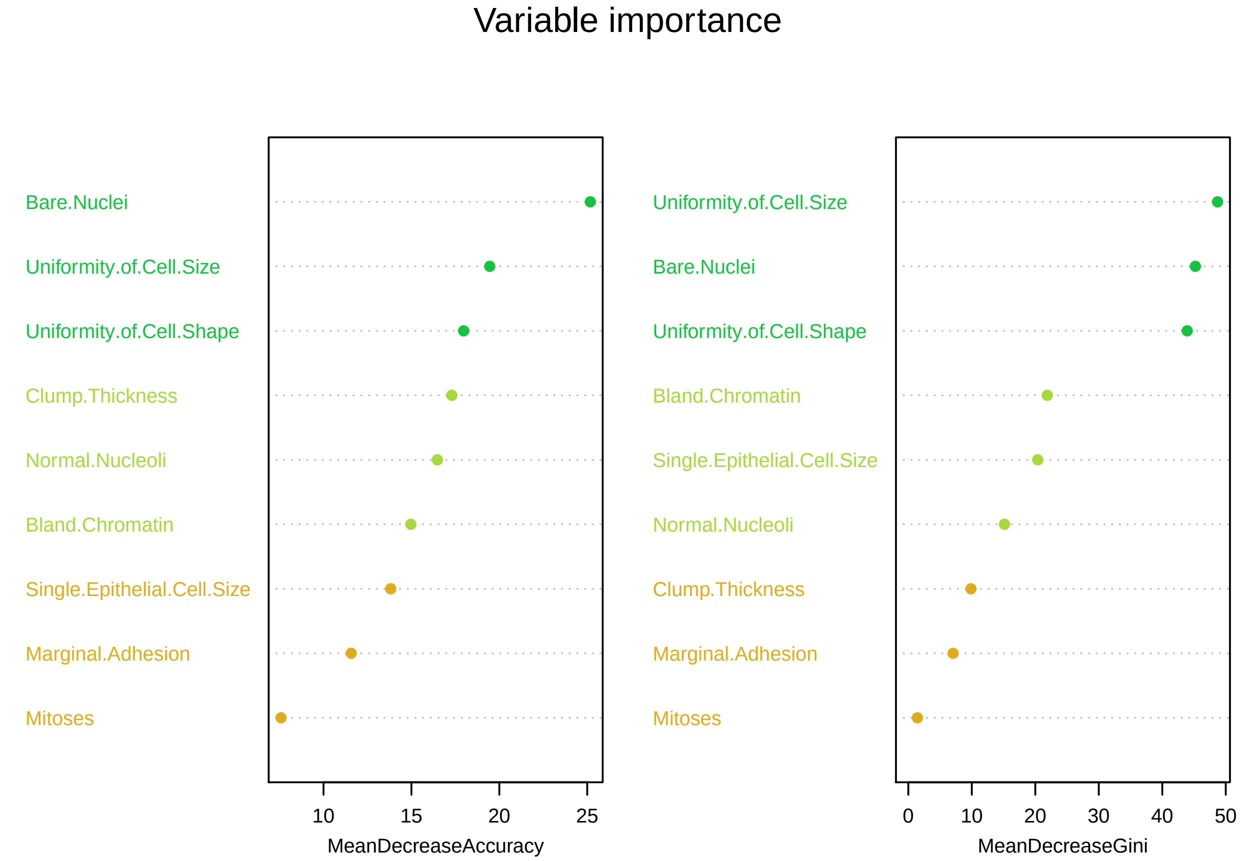
If we consider the sum of importance of all the variables, the part of it intended for `Mitoses` and `Marginal.Adhesion` are very low and equal:

```
## [1] 0
## [1] 0.022567
```

which is definitely very low.

4.1.3 Breiman's random forest algorithm

We will consider `randomForest` function from `randomForest` package.



Once again, both `Mitoses` and `Marginal.Adhesion` resulted being the least important. `Bare.Nuclei` for the second time showed its importance. Also two variables describing the unformity of the cell are relevant. We have mixed feeling to the remaining features as being classified differently for both mean decrease in accuracy and mean decrease in impurity calculated using Gini index.

4.1.4 Stepwise feature selection

As the last method, we will perform a stepwise feature selection using function `stepAIC()` from `MASS` package based on linear regression model. The function begins with null or full model and in each step it respectively adds the most significant feature or removes the least significant one, looking for minimum value of Akaike Information Criterion. Of course, here we need to treat our data as numerical. The summary of full model is as follows:

```r
linear.model1 <- lm(Class ~ ., data = train_num)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.2524	0.0200	-12.61	0.0000
Clump.Thickness	0.0303	0.0043	7.10	0.0000
Uniformity.of.Cell.Size	0.0190	0.0075	2.52	0.0122
Uniformity.of.Cell.Shape	0.0099	0.0075	1.31	0.1902
Marginal.Adhesion	0.0078	0.0050	1.57	0.1181
Single.Epithelial.Cell.Size	0.0072	0.0062	1.17	0.2429
Bare.Nuclei	0.0485	0.0040	12.04	0.0000
Bland.Chromatin	0.0209	0.0061	3.40	0.0007
Normal.Nucleoli	0.0232	0.0044	5.33	0.0000
Mitoses	0.0132	0.0062	2.13	0.0338

Table 3: Summary of a full model constructed with lm function.

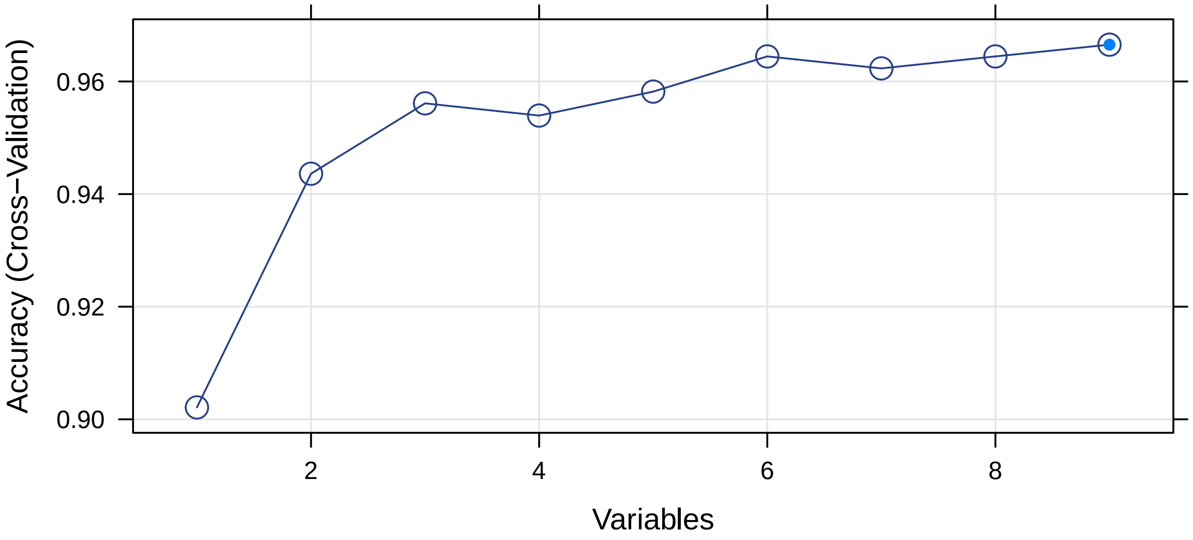
This summary already suggest very little importance of `Single.Epithelial.Cell.Size` and using backward, forward and both-sides selection only confirms that. This method does not necessarily mean to improve the model performance, but it is used to simplify the model without impacting much on the performance and loosing important information. Also the second highest p-value can be noticed for `Uniformity.of.Cell.Shape` and it is probably the reason of very strong correlation between the shape and size of cell uniformity which is equal to 0.91. It confirms our assumptions that we can omit one of those two variables.

In package `klaR` we can find another interesting function, `stepclass()`. It is used for forward/backward variable selection for classification using any specified classification function. We will check its results for LDA and QDA methods using forward direction. We set stop criterion if improvement is less than 0.1%

```r
lda.selection <- stepclass(Class ~ ., data=train_num, method="lda",
                           direction="forward", improvement=0.001)
qda.selection <- stepclass(Class ~ ., data=train_num, method="qda",
                           direction="forward", improvement=0.001)
```

The first function returns the following variables (in importance order):
`Bare.Nuclei`, `Uniformity.of.Cell.Size` and `Clump.Thickness`.
The second function returns:
`Uniformity.of.Cell.Size`, `Bare.Nuclei`, `Clump.Thickness` and `Uniformity.of.Cell.Shape`.
As we can see, they differ a little bit for each method. However, `Bare.Nuclei` and `Uniformity.of.Cell.Size` have again the biggest importance.

Another algorithm worth looking at is recursive feature elimination (`rfe()` function). In other words it is called backwards feature selection. We can find it in `caret` package. After a proper ranking, the less relevant predictors are eliminated prior to modeling.



The plot shows that selecting just three variables is enough to get very high accuracy.

```
##
## Recursive feature selection
##
## Outer resampling method: Cross-Validated (10 fold)
##
## Resampling performance over subset size:
##
##  Variables Accuracy  Kappa AccuracySD KappaSD Selected
##          1   0.9021 0.7818    0.06003 0.13587
##          2   0.9436 0.8760    0.03261 0.07205
##          3   0.9561 0.9039    0.03197 0.06867
##          4   0.9539 0.8990    0.02392 0.05140
##          5   0.9582 0.9078    0.02201 0.04813
##          6   0.9645 0.9216    0.02207 0.04840
##          7   0.9623 0.9171    0.02378 0.05221
##          8   0.9645 0.9226    0.02607 0.05573
##          9   0.9665 0.9269    0.02446 0.05233        *
##
## The top 5 variables (out of 9):
##    Bare.Nuclei, Uniformity.of.Cell.Size, Clump.Thickness, Uniformity.of.Cell.Shape, Bland.Chromatin
```

We can see that te function returns the list of 5 variables that most influences the model. The top is taken by `Bare.Nuclei` and `Uniformity.of.Cell.Size` again.

One feature marked with a star was eliminated using this algorithm. It is not a surprise that this feature is `Mitoses`, which we have expected to be the least relevant for quite some time.

4.1.5 Feature selection conclusion

We have checked multiple methods of feature selection. While Boruta and Stepwise told us exactly which features to treat as important, other methods provided the ranking of variables. Our dataset contains 9 variables, which is not a lot. However, all approaches to feature selection stated that `Mitoses` is the least relevant one. Moreover, `Marginal.Adhesion` showed disappointing values of importance related to other features. Also `Uniformity.of.Cell.Shape` is strongly correlated with `Uniformity.of.Cell.Size`, so removing it will not deteriorate the model significantly. The remaining features of breast cancer were located in different positions when it comes to importance. In our further actions we will check the classification accuracy for a couple of sets — the complete one consisting of 9 variables and the others consisting of a few selected variables, which are the best for particular model.

4.2 Dealing with class imbalance problem

As noticed before, there is a slight class imbalance in our data set. The majority of examined women suffer from the benign cancer. Only 35% of women had to cope with malignant one. We could deal with that problem using `SMOTE` function from `DMwR` package.

```
balanced.train <- SMOTE(Class ~., data = train, perc.over = 250,
                        perc.under = 150)
```

The new balanced data set is divided into classes equally:

```
##
##    Benign Malignant
##       486       486
```

We have tested the equally divided data set. There was no huge improvement in the classification results. As we mentioned before, the imbalance problem for our dataset is not a glaring issue and we are able to answer all the questions stated at the beginning without balancing the data set. Moreover, the number of informations we do have for malignant cancer is fairly sufficient, so let's have a look at the further classification analysis.

4.3 Linear regression model

First of all, we will try to build a linear regression model. The prediction of the dependent variable Y, which is a type of cancer in our case, is given by the formula:

$$\hat{Y} = \underline{X}^T \underline{\hat{\beta}},$$

where $\underline{X}^T = (1, X_1, ..., X_p)$, $\underline{\hat{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_p)^T$ and p is a number of independent variables.

The least squares method solution to estimate unknown coefficients is given by:

$$\underline{\hat{\beta}} = (\mathbb{X}^T \mathbb{X})^{-1} \mathbb{X}^T \underline{y},$$

where $\mathbb{X}_{n \times (p+1)}$ is a model matrix and $\underline{y} = (y_1, y_2, ..., y_n)^T$ is a vector containing dependent variable ($y_i \in \{0, 1\}$).

Since we are not dealing with continuous data, we have to code our classes as `Benign` $= 0$, `Malignant` $= 1$ and set a classification rule

$$\hat{G} = \begin{cases} \text{Benign}, & \text{if } \hat{Y} \leq 0.5 \\ \text{Malignant}, & \text{if } \hat{Y} > 0.5 \end{cases} \tag{1}$$

We will compare the performance of a full model and a model basing on all variables except for `Mitoses`, `Uniformity.of.Cell.Shape` and `Marginal.Adhesion`.

```
lm.all <- lm(Class ~ ., train_num)
pred.all <- ifelse(predict(lm.all, test_num) > 0.5, 1, 0)
```

The remaining models are built similarly. Let us consider the whole dataset at a starting point. The confusion matrix looks like this:

	0	1
0	126	2
1	7	70

while 0.04390244 is the misclassification error.

```
lm.sel8 <- lm(Class ~ . - Mitoses, train_num)
pred.sel8 <- ifelse(predict(lm.sel8, test_num) > 0.5, 1, 0)
```

After removing `Mitoses` the confusion matrix is following:

	0	1
0	126	2
1	6	71

and 0.03902439 is the value of error.

```
lm.sel7 <- lm(Class ~ . - Mitoses - Marginal.Adhesion, train_num)
pred.sel7 <- ifelse(predict(lm.sel7, test_num) > 0.5, 1, 0)
```

Getting rid of `Marginal.Adhesion` additionally, resulted in the value 0.03902439 of misclassification error, so it has not even changed. It means that we can remove those two variables and sleep peacefully after that. The confusion matrix is presented below.

	0	1
0	126	2
1	6	71

```
lm.sel6 <- lm(Class ~ . - Mitoses - Marginal.Adhesion - Uniformity.of.Cell.Shape, train_num)
pred.sel6 <- ifelse(predict(lm.sel6, test_num) > 0.5, 1, 0)
```

For the independent variables set containing 6 variables (`Mitoses`, `Marginal.Adhesion`, `Uniformity.of.Cell.Shape` excluded), the confusion matrix is shown below:

	0	1
0	126	2
1	6	71

The misclassification error is then 0.03902439 and it is also the same.

We will stop here, since removing another feature decreases the accuracy.

After this quick revision of a few selected models we can see that the prediction model can be simplified by decreasing number of component features even by three, keeping the same goodness of fit. For this particular division into train and test sets the misclassification error is about 3.9%. We will check whether different algorithms give better results.

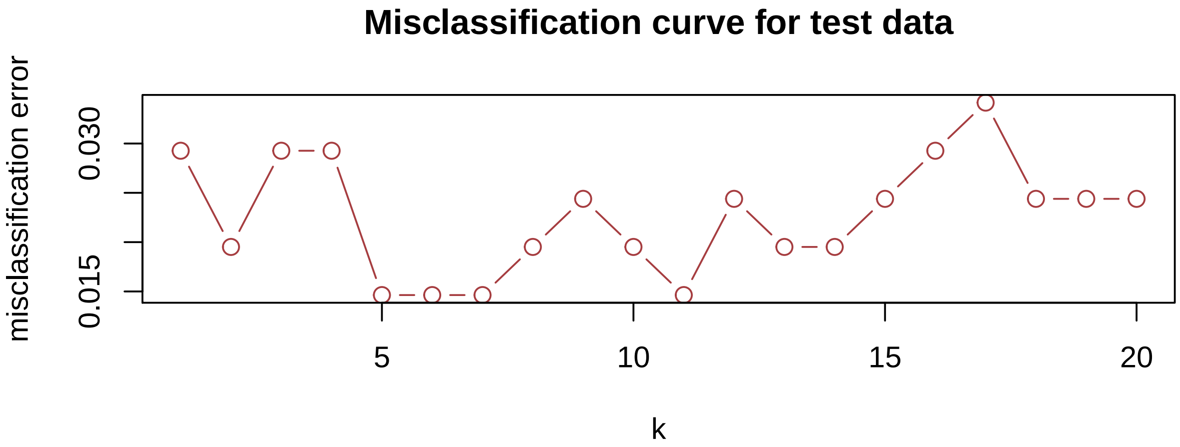
4.4 k nearest neighbours algorithm

k-NN algorithm is a simple supervised machine learning algorithm used to solve classification problems. It is based on a distance between two points. For this case we will assume that we can measure the distance between two values as in Euclidean space.
Analogously to regression problem, we have classes 0 and 1, the same classification rule, but the prediction formula is as follows:

$$\hat{Y}(\underline{x}) = \frac{1}{k} \sum_{\underline{x}_i \in N_k(\underline{x})} y_i,$$

where $N_k(\underline{x})$ is k nearest neighbours for observation $\underline{x}$ in the training set.

In order to select the optimal k parameter, we check the misclassification error for a range of its values.



The smallest error is obtained for $k = 5$ (±1 for this dataset division), so we will use this value for further models comparison. Again we will build a full model and then a few simpler ones.

```
knn.all <- ipredknn(Class ~ ., data=train_num, k=5)
pred.all <- predict(knn.all, test_num, type="class")
```

Let's have a look at the confusion matrix and misclassification error: 0.01463415

	0	1
0	126	2
1	1	76

The following two models presented are different. The first one contains 7 variables (the whole data set excluding `Mitoses` and `Marginal.Adhesion`) and the second one consists of only 4 variables suggested in previous part of the analysis: `Bare.Nuclei`, `Clump.Thickness`, `Uniformity.of.Cell.Size` and `Uniformity.of.Cell.Shape`.

```
# for 7 selected features
knn.sel7 <- ipredknn(Class ~ . - Mitoses - Marginal.Adhesion, data=train_num, k=5)
```

The misclassification error: 0.01463415

	0	1
0	125	3
1	0	77

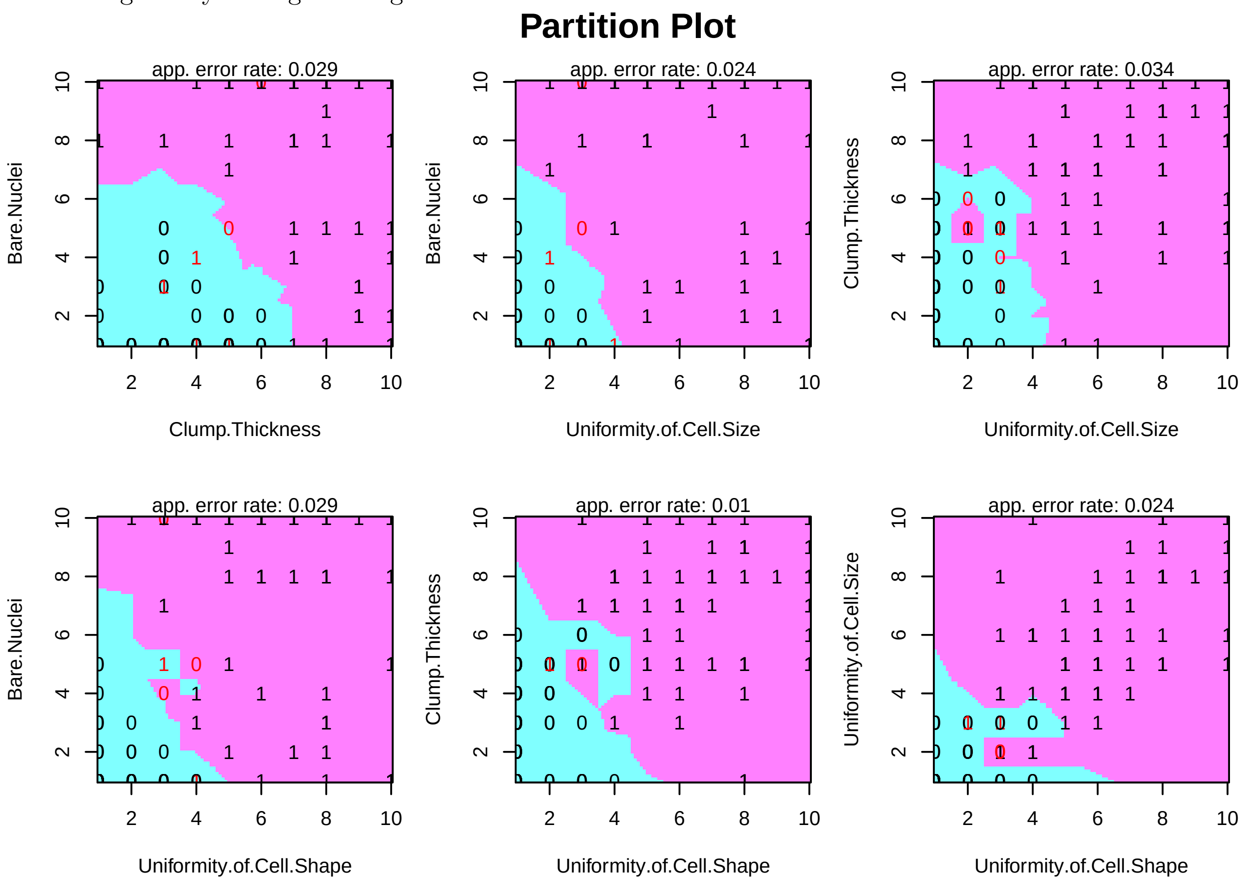
```
# for 4 selected features
knn.sel <- ipredknn(Class ~ Bare.Nuclei + Clump.Thickness + Uniformity.of.Cell.Size
                     +  Uniformity.of.Cell.Shape, data=train_num, k=5)
```

The misclassification error: 0.0195122

	0	1
0	125	3
1	1	76

Excluding `Mitoses` and `Marginal.Adhesion` from k-NN analysis has not increased the number of misclassified observations. However, this approach gives different results for each repetition (due to the random order of selecting points in the algorithm), although the k-NN model consisting of only 4 variables still gives better result than the best linear regression model.

If we take a look at the decision boundaries for all pairs of variables for a model containing only `Bare.Nuclei`, `Clump.Thickness`, `Uniformity.of.Cell.Size` and `Uniformity.of.Cell.Shape`, we cannot see any nice boundary. Thus, it is consistent with our preliminary analysis, that it is very hard to talk about any discriminating ability among our single attributes.



4.5 Linear Discriminant Analysis

LDA approach bases on the assumption of multivariate normality and homogeneity of variance/covariance. It is used in general for datasets with continuous independent variables and categorical dependent variable, so again we will treat our data as numeric. The decision rule construction starts with considering log-odds ratio. In such a case the boundaries of decision regions are linear. For binary classification problem, if the number of observations in each class is equal, classification based on the LDA is equivalent linear regression classification. It is not satisfied for our dataset, thus we will prepare new models using `lda()` function from `MASS` package and compare the results.

```
lda.all <- lda(Class ~ ., data=train_num)
```

The misclassification: 0.03902439

	0	1
0	126	2
1	6	71

```
# for 3 selected features according to forward selection
lda.sel <- lda(Class ~ Clump.Thickness + Uniformity.of.Cell.Size +
                 Bare.Nuclei, data=train_num)
```

The misclassification: 0.02926829

	0	1
0	127	1
1	5	72

For a given training/testing dataset division the full LDA model returns similar matrix to the full regression model. The second proposed model consists of only three features, which were chosen in the previous section as the most important. Here we got two less misclassified observations, which means that we managed to simplify much the model and not loose any important information, and even boost its performance. Of course we keep in mind that it also depends on the training and testing set division and cross validation might give more reliable results.

4.6 Quadratic Discriminant Analysis

QDA is generalized version of LDA, where the assumption of equal variance/covariance matrices is not needed. Again we will analyze full model and the model with variables selected according to forward stepwise analysis from previous section.

```
qda.all <- qda(Class ~ ., data=train_num)
```

The misclassification error: 0.02439024

	0	1
0	124	4
1	1	76

```
# for 4 selected features according to forward selection
qda.sel <- qda(Class ~ Clump.Thickness + Uniformity.of.Cell.Size +
                 Uniformity.of.Cell.Shape + Bare.Nuclei, data=train_num)
```

The misclassification error: 0.01463415

	0	1
0	126	2
1	1	76

We managed to build much simpler classification model keeping the accuracy at a very high level. Comparing to LDA, QDA performed here much better.

4.7 Logistic regression

We will use function glm from stats package to perform logistic regression. Independent variables are treated as numeric. Since the classes are coded as 0 and 1, we can easily transform our prediction response (posterior probability) into predicted classes by comparing the probability with a given cutoff level. We will assign the observation with response larger than 0.5 to class 1 and the rest to class 0.

```r
logit.all <- glm(Class ~ ., data=train_num, family=binomial(link="logit"))
pred.all <- ifelse(predict(logit.all, test_num, type = "response") > 0.5, 1, 0)
```

The misclassification error: 0.0195122

	0	1
0	126	2
1	2	75

```r
# for 4 selected features
logit.sel <- glm(Class ~ Bare.Nuclei + Clump.Thickness +
                   Uniformity.of.Cell.Size + Uniformity.of.Cell.Shape, data=train_num,
                   family=binomial(link="logit"))
```

The misclassification error: 0.0195122

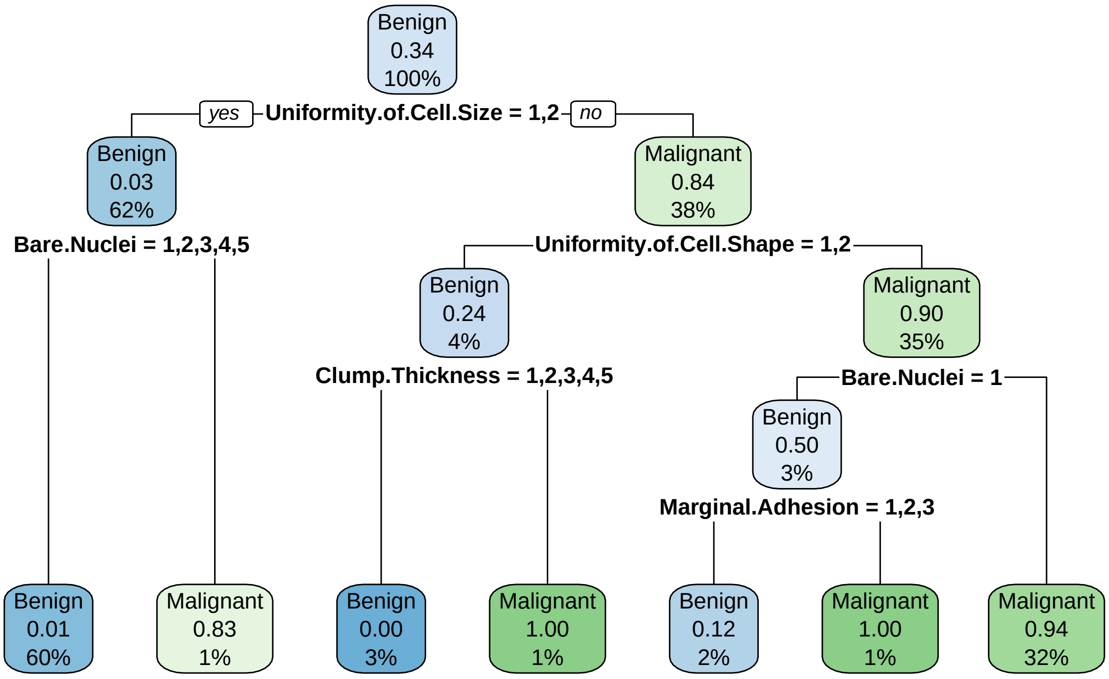
	0	1
0	126	2
1	2	75

Again four variables, Bare.Nuclei, Clump.Thickness, Uniformity.of.Cell.Size, Uniformity.of.Cell.Shape were sufficient to construct a good classification model. Accuracy is comparable with k-NN method's one and better than LDA's one (probably because it imposes fewer assumptions on data).

4.8 Classification tree

We are going to use a binary tree model on our dataset. Based on selected attributes' values, we will split the data into partitions. The function `rpart` from `rpart` package will do the job. In case of decision trees we do not need to specify most important variables manually — the algorithm does it on its own. So we start with building a complex tree and then we prune it choosing optimal `cp` parameter. Let's now have a look at the tree below. In accordance with our first though that smaller values of features indicates the benign cancer, that's how the splits are made.

Pruned Classification Tree − the whole dataset Breast Cancer



The confusion matrix for class prediction is following:

	Benign	Malignant
Benign	125	3
Malignant	7	70

One binary tree is so far the worst classifier for our Breast Cancer data. But let's now get ready for the better version of it — the forest consisting of many trees.

4.9 Random forest

Random forest based on Breiman's random forest algorithm is one of the ensemble methods. They are based on averaging or majority voting. So the ensemble-based classification is defined as the combined (or aggregated) classification, which makes decisions based on the components. The results are more reliable as based on many components, not just one. Random forest shows excellent results comparable with the best-known classifiers. It does not overfit. Random forest can also handle the variable selection, but we have digged through it in feature selection section. For our analysis we will compare forests containing different number of trees.

For 6 trees in our forest the results are following:

```r
p <-  ncol(train) - 1
rf1 <- randomForest(Class~., data=train, ntree=6, mtry=round(sqrt(p)), importance=TRUE)
```

	Benign	Malignant
Benign	125	2
Malignant	3	75

On the other hand, for 120 trees we obtain:

```r
p <-  ncol(train) - 1
rf2 <- randomForest(Class~., data=train, ntree=120, mtry=round(sqrt(p)), importance=TRUE)
```

	Benign	Malignant
Benign	125	0
Malignant	3	77

Is it possible to get a better accuracy? Let's have a look at the forest containing 700 trees:

```r
p <-  ncol(train) - 1
rf3 <- randomForest(Class~., data=train, ntree=700, mtry=round(sqrt(p)), importance=TRUE)
```

	Benign	Malignant
Benign	125	0
Malignant	3	77

As we can see, 120 already forms a good random forest. Even 6 trees gave much better results than 1 single tree.

5 Deeper analysis of selected methods

In this part we will take a closer look on the models which gave the best results in the previous section. We will perform a 5-fold cross validation (on a shuffled data frame) to avoid the consequences of dependence on training subset selection, and inspect a few more accuracy measures.

We will skip the feature selection process in each iteration and set, in advance, the most important variables, basing on our past experiance. Three models will be compared:

- full model, 9 independent variables,
- simplified model built on 7 independent variables (all except for `Mitoses` and `Marginal.Adhesion`)
- simplified model built on 3 independent variables (`Bare.Nuclei`, `Clump.Thickness` and `Uniformity.of.Cell.Size`)

5.1 k nearest neighbours

Firstly, we will compare average misclassification error, sensitivity, specificity and precision for all three mentioned above models.

	Mean	Variance
Misclassification error	0.0264	0.00058
Sensitivity	0.9657	0.00118
Specificity	0.9779	0.00045
Precision	0.9572	0.00207

Table 4: Accuracy measures for full model.

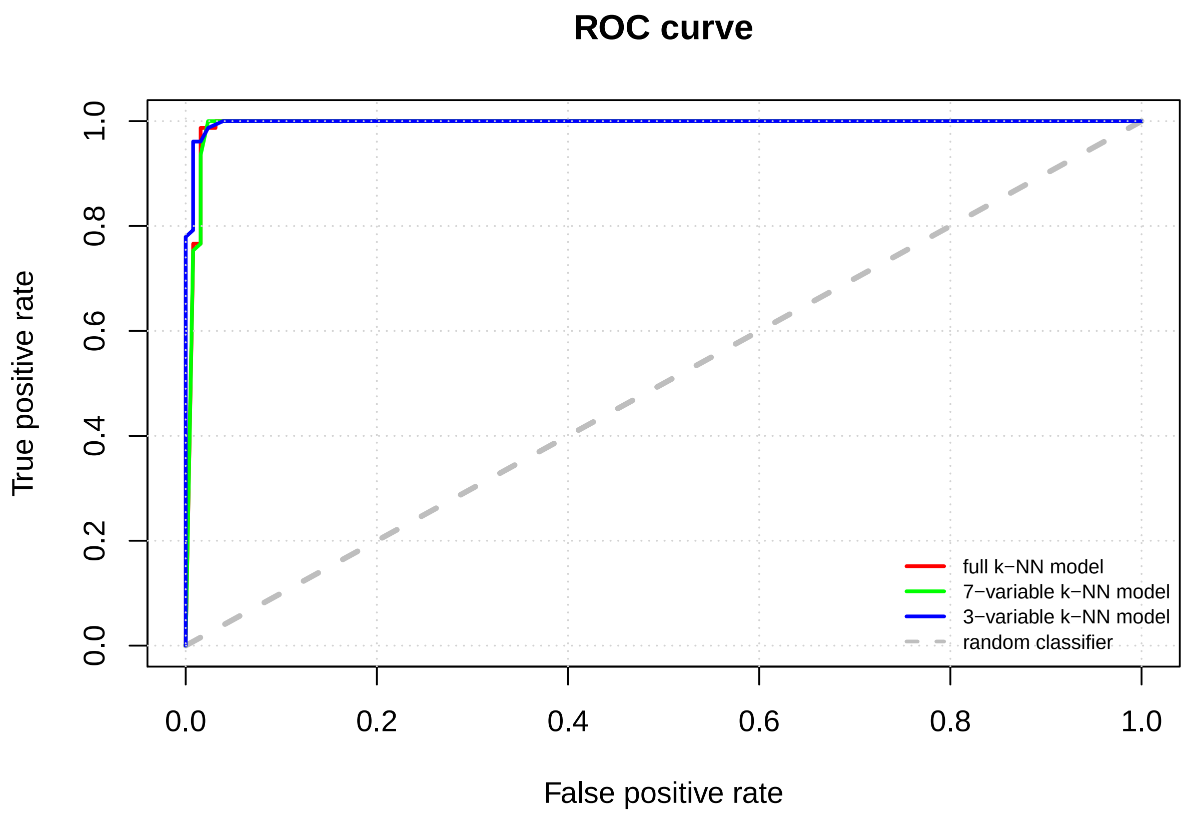
	Mean	Variance
Misclassification error	0.0264	0.00031
Sensitivity	0.9706	0.00012
Specificity	0.9757	0.00063
Precision	0.9543	0.00270

Table 5: Accuracy measures for simplified model (7 variables.

	Mean	Variance
Misclassification error	0.0337	0.00034
Sensitivity	0.9625	0.00056
Specificity	0.9687	0.00039
Precision	0.9417	0.00137

Table 6: Accuracy measures for simplified model (3 variables).

Very little variance means that there is no big difference in accuracy measures for different subset divisions.



Excluding 2 variables almost did not affect the quality of prediction. Excluding 5 variables caused slight deterioration in accuracy, but the simplification is significant. For the future medical research simplification can prevent from overfitting and suggest the class membership basing on smaller number of measurments. In addition, area under ROC curve in all cases is really close to 1 and the difference between them is very small, so we are satisfied with our simplified k-NN models.

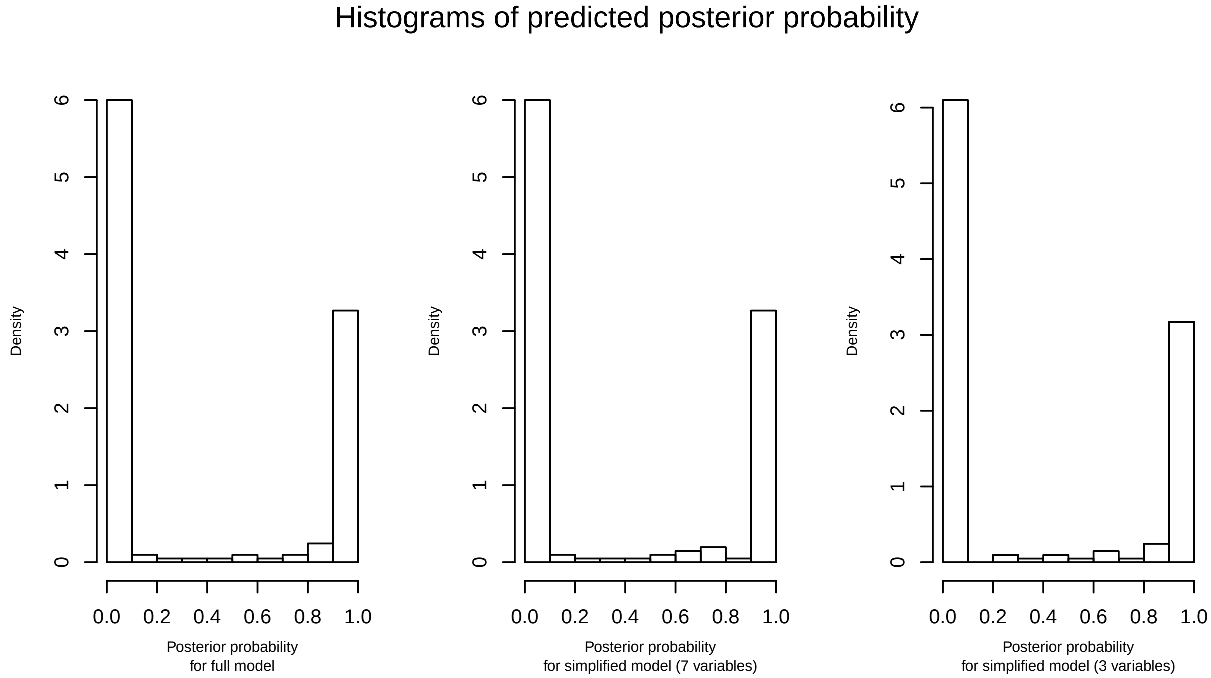
5.2 Logistic regression

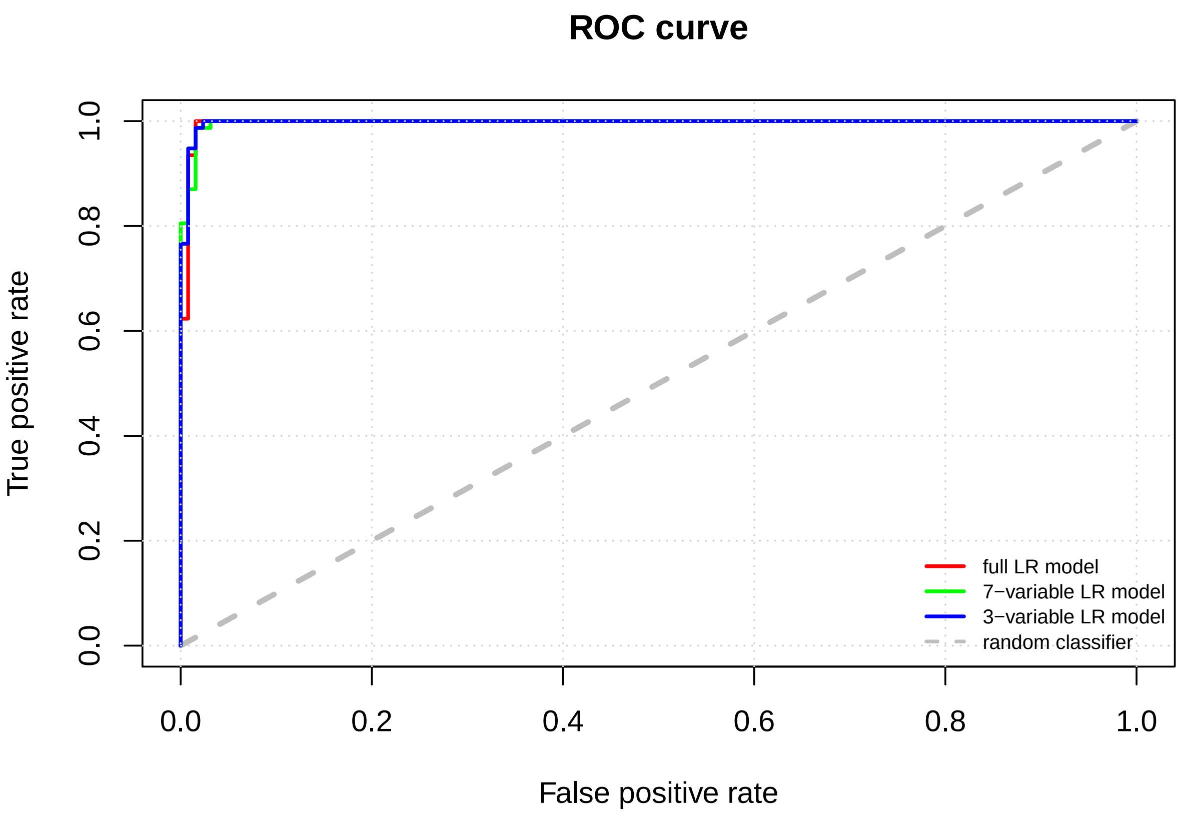
Logistic regression was the second method which gave the best result in previous analysis. For one given training/testing subset division the full model ranks Clump.Thickness and Bare.Nuclei as the most important ones.

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-9.6489	1.2947	-7.45	0.0000
Clump.Thickness	0.5058	0.1542	3.28	0.0010
Uniformity.of.Cell.Size	0.0062	0.2669	0.02	0.9814
Uniformity.of.Cell.Shape	0.1027	0.2837	0.36	0.7174
Marginal.Adhesion	0.3200	0.1455	2.20	0.0278
Single.Epithelial.Cell.Size	0.0614	0.1795	0.34	0.7321
Bare.Nuclei	0.4650	0.1189	3.91	0.0001
Bland.Chromatin	0.4008	0.1949	2.06	0.0397
Normal.Nucleoli	0.3074	0.1353	2.27	0.0231
Mitoses	0.6260	0.3050	2.05	0.0401

Table 7: Summary of a full model constructed with glm function.

Histograms of predicted posterior probabilities (for given training/testing subset division) for all three cases show that the algorithm is pretty sure about class assignment and manipulating cutoff level will not affect the result much. Thus we will keep the cutoff level 0.5.





The ROC curves indicate very good performance of all models. At the end we will perform 5-fold cross validation to compare the average accuracy measurments.

	Mean	Variance
Misclassification error	0.0337	0.00028
Sensitivity	0.9493	0.00061
Specificity	0.9756	0.00019
Precision	0.9521	0.00100

Table 8: Accuracy measures for full model.

	Mean	Variance
Misclassification error	0.0322	0.00044
Sensitivity	0.9495	0.00121
Specificity	0.9778	0.00022
Precision	0.9561	0.00115

Table 9: Accuracy measures for simplified model (7 variables).

	Mean	Variance
Misclassification error	0.0351	0.00017
Sensitivity	0.9432	0.00049
Specificity	0.9777	0.00028
Precision	0.9572	0.00127

Table 10: Accuracy measures for simplified model (3 variables).

The final conclusion about LR models is analogous to the one about k-NN models. The restriction to 3 independent variables enables to make the model more general not loosing much accuracy in the same time.

In final comparison, the best classificator for Breast Cancer dataset turns out to be k-NN algorithm with $k = 5$.

6 Conclusions and remarks

The analysis helped us answer all the questions we have asked at the beginning and dispelled all our doubts. Most of the methods we discussed were applied to numerical data. We kept in mind that the values in our data frame were actually categorical, but that worked very well. Only classification tree and random forest worked directly on data of type factor. After the "first shot" classification, we chose k-NN algorithm and logistic regression for closer insight. Carrying out cross validation assured us about their good performance. We decided not to repeat feature selection in each iteration in favour of generalizing the importance for all measures and making model independent of training/testing subset division. It turned out that three variables are sufficient to predict the class of tumor with good accuracy. However, we should consider the purposes of the whole analysis and classification. If it is supposed to be used for only statistical purposes, having information about Bare Nuclei, Clump Thickness and Uniformity of Cell Size will give us satisfactory results. When it comes to patient's diagnosis, we should be as accurate as possible, so rather use full models or models excluding Mitoses and Marginal Adhesion, since our analysis proved their negligible importance. Giving less attention to type I and type II errors might result with strongly confusing diagnoses, which might have fatal consequences.

7 Bibliography

References

[1] Medical diagnostics: Breast Cancer Wisconsin (Original) Data Set
`http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original).`
UCI, Machine Learning Repository, 1992.

[2] H. Liu, H. Motoda, *Computational Methods of Feature Selection* (2008)

[3] Carolin Strobl (LMU Muenchen) and Achim Zeileis (WU Wien), *Why and how to use random forest variable importance measures* (2008)

[4] Yvan Saeys, Inaki Inza, Pedro Larranaga, *A review of feature selection techniques in bioinformatics* (2007)

[5] Feature Selection – Ten Effective Techniques, `www.machinelearningplus.com` (2017)

[6] Dr. Bharatendra Rai, *Feature Selection using R* (2018)

[7] A. Zagdanski, Lecture materials for Data Mining course (2019)