Joanna Matuszak 255762, Joanna Wojciechowicz 255747

Microarray brain tumor diagnostics

February 6, 2024

Contents

1. Introduction

In this project, we introduce exploratory data analysis and multiclass classification on microarray data regarding brain tumors. Microarray data experiments monitor gene expression in different tissues. The experiment is equipped with an additional response variable such as a cancer type. In this dataset, we have 5 different tumor types in the response variable - class. Furthermore, the number of measured genes is more than 5 thousand. It is assumed that only a few marker components of this gene subset determine the type of tissue. The identification of these significant groups is crucial for tumor classification in medical diagnostics, as well as for understanding how the genome as a whole works.

Accordingly, before exploratory data analysis, we performed feature selection to focus only on the most meaningful features for the whole dataset. The feature selection procedure and its stability will be described in the following part of the project. In the analysis, we took into account various aspects of the data like distribution in classes, different statistics, discriminative ability of each feature, correlation between chosen genes, data visualization, etc. Based on the knowledge that we gained in this part, we could move to the classification problem.

In the classification part, we took into account various classification algorithms: K Nearest Neighbors, Linear and Quadratic Discriminant Analysis, Multinomial Logistic Regression, Decision Tree and Random Forest. Since we faced a problem of a large number of features and a very small number of observations ($p \gg n$), we incorporated a feature selection procedure into our model selection pipeline. We performed leave-one-out cross-validation on the algorithms to extract the best parameters for our models. Finally, we compared our models with chosen parameters based on misclassification errors and macro-averaged F1 score. We also investigated their stability using box plots. At the end, we suggested further research directions that might improve our work.

The main research problem is the evaluation of the chosen feature selection method based on the comparison of models' results with feature selection and with randomly chosen features. Another important problem is the selection of the best model for our research, investigating the stability of the feature selection procedure, and the stability of chosen models. This study might yield a better diagnostic method, where doctors might get suggestions on tumor type from the model.

2. Feature preselection

In the brain tumor data set we face a problem of large number of genes. The sample size n is much smaller than the dimensionality of the feature space - the number of genes p. We are looking for the most important features, with strong discriminant ability. It drastically reduces computational time and might improve the results - due to reduced amount of noise.

In feature selection procedure[1] we score each gene $g \in \{1, \ldots, p\}$, according to its strength for phenotype discrimination. We base our approach on Wilcoxon's two sample test,

$$s(g) = \sum_{i \in N_0} \sum_{j \in N_1} 1_{[x_j^{(g)} - x_i^{(g)} \leq 0]},$$

where $x_i^{(g)}$ is the expression value of gene g for individual i and N_m represents the set of the n_m indices, where $n_m \in \{1, \ldots, n\}$, having response in $m \in \{0, 1\}$. It is easy to notice that both values near the minimum score zero and the maximum score $n_0 n_1$ indicate strongly informative gene. It is because we can interpret the score function as counting for each individual having response value zero, the number of instances with response one that have smaller expression values. We introduce quality measure

$$q(g) = max\left(s(g), n_0 n_1 - s(g)\right),$$

which gives the highest values to those genes that express the strongest discriminative ability in terms of phenotype discrimination.

This procedure is implemented in the code below.

```
# Wilcoxon's test
score <- function(x, y, flag)
{
  if (flag == 'statistic') {
    wilcox.test(x[which(y==0)], x[which(y==1)])$statistic
  }
  else if (flag == 'p.value') {
    wilcox.test(x[which(y==0)], x[which(y==1)])$p.value
  }
}

# Best features based on given learning set
feature.preselection <- function(xlearn, ylearn, presel = 40)
{
  s  <-  apply(xlearn, 2, score, ylearn, 'statistic')
```

```
  a <- (sum(ylearn==0)*sum(ylearn==1)) - s
  quality <- apply(rbind(s,a),2,max)
  genes   <- rev(order(quality))[1:presel]
  qualities <- quality[genes]
  best.features <- list("genes"=genes, "qualities"=qualities)
  return(best.features)
}
```

3. Exploratory data analysis on chosen features

In our brain tumor microarray data set, we have:
- 5598 features (including label),
- 42 observations,
- 5 classes in the target variable - $0, 1, 2, 3, 4$.

All 5597 variables are quantitative continuous features, and the target variable - class - is discrete nominal. We do not have any missing values or duplicated observations in our data.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	Modal interval
V4080	0.43	0.73	0.97	1.01	1.26	1.72	(0.9 ,1.1)
V4163	-0.5	0.15	0.45	0.43	0.77	1.15	(0.7 ,0.9)
V1892	-1.57	-0.98	-0.09	0.06	0.83	2.21	(-1.25 ,-0.75)
V1033	0.59	0.96	1.23	1.22	1.47	1.79	(1.3 ,1.5)
V3815	-0.05	0.97	1.36	1.31	1.78	2.16	(1.75 ,2.25)
V3338	-1.49	0.23	0.56	0.43	0.78	1.82	(0.75 ,1.25)
V845	-1.79	-1.42	0.07	-0.25	0.42	1.52	(0.25 ,0.75)
V1394	-1.73	-1.2	-0.53	-0.62	-0.06	1.23	(-0.25 ,0.25)
V2848	-1.57	0.02	0.4	0.32	0.66	1.52	(0.75 ,1.25)
V2445	-1.75	-0.92	-0.41	-0.46	0.03	0.46	(-0.25 ,0.25)
V2429	-0.26	0.32	0.63	0.6	0.91	1.41	(0.7 ,0.9)
V540	-1.81	-1.6	-1.44	-0.93	-0.17	0.72	(-1.75 ,-1.25)
V4274	-1.79	-1.46	-0.8	-0.88	-0.49	0.53	(-0.75 ,-0.25)
V4318	-1.6	-0.93	-0.52	-0.46	-0.1	1.17	(-0.75 ,-0.25)
V4356	-0.13	0.43	0.61	0.63	0.79	1.41	(0.7 ,0.9)
V4589	-1.52	0.15	0.49	0.41	0.75	1.55	(0.25 ,0.75)
V1367	-1.62	-0.7	0.1	-0.1	0.53	1.44	(0.25 ,0.75)
V5546	-1.89	-1.48	-1.07	-0.94	-0.42	0.38	(-1.25 ,-0.75)
V808	-1.75	-1.49	-0.52	-0.61	0.06	0.73	(-1.75 ,-1.25)
V5316	-1.81	-1.56	-0.73	-0.72	0.06	0.61	(-1.75 ,-1.25)

Table 3.1. Summary matrix of selected features

On the graph depicting class imbalance (3.1), we can see that we have an imbalanced classes problem in our dataset. Class 3 and class 4 have significantly fewer observations than the rest of the classes.

We face a problem of a large number of features versus a small number of observations ($p \gg n$). In exploratory data analysis, we will use the already described feature selection approach to extract the most important features for the entire dataset. Since feature selection works in a one-vs-all manner,

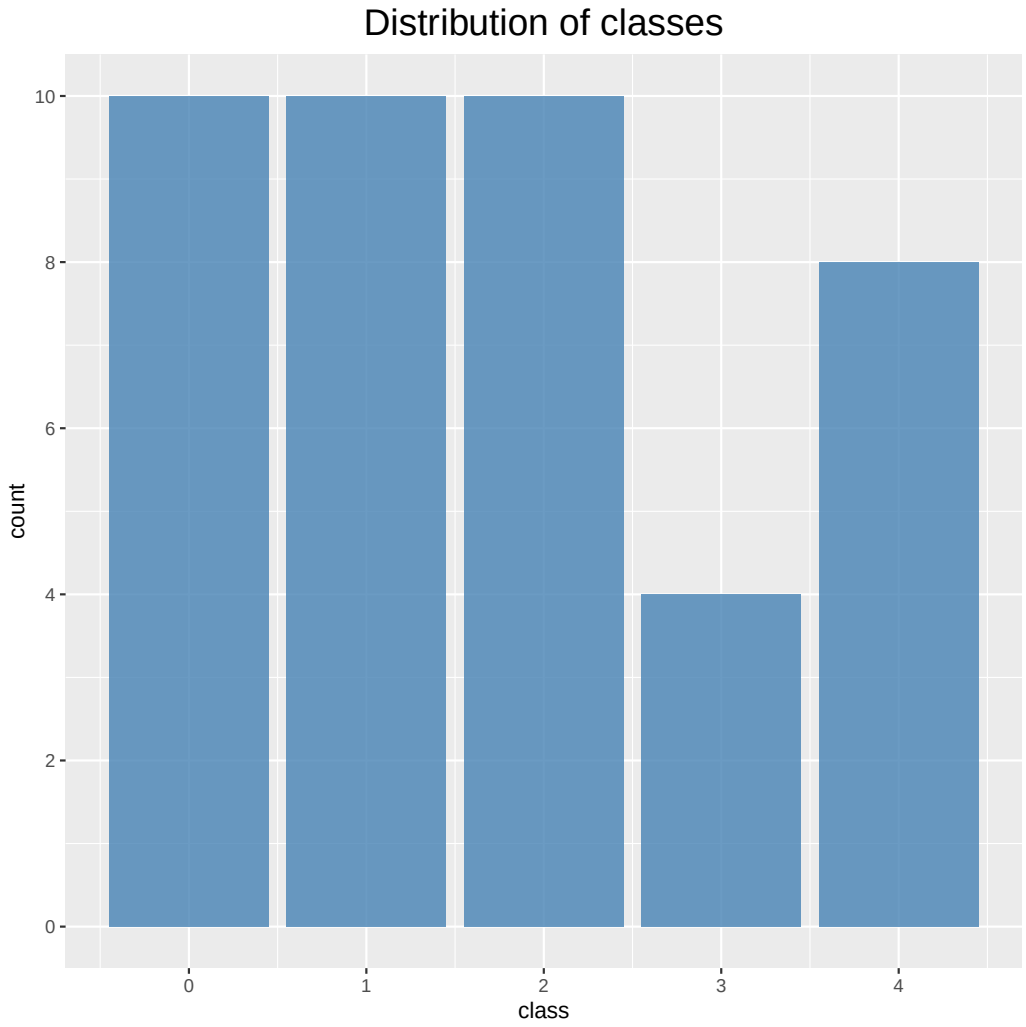

Figure 3.1. Distribution of each class in the dataset

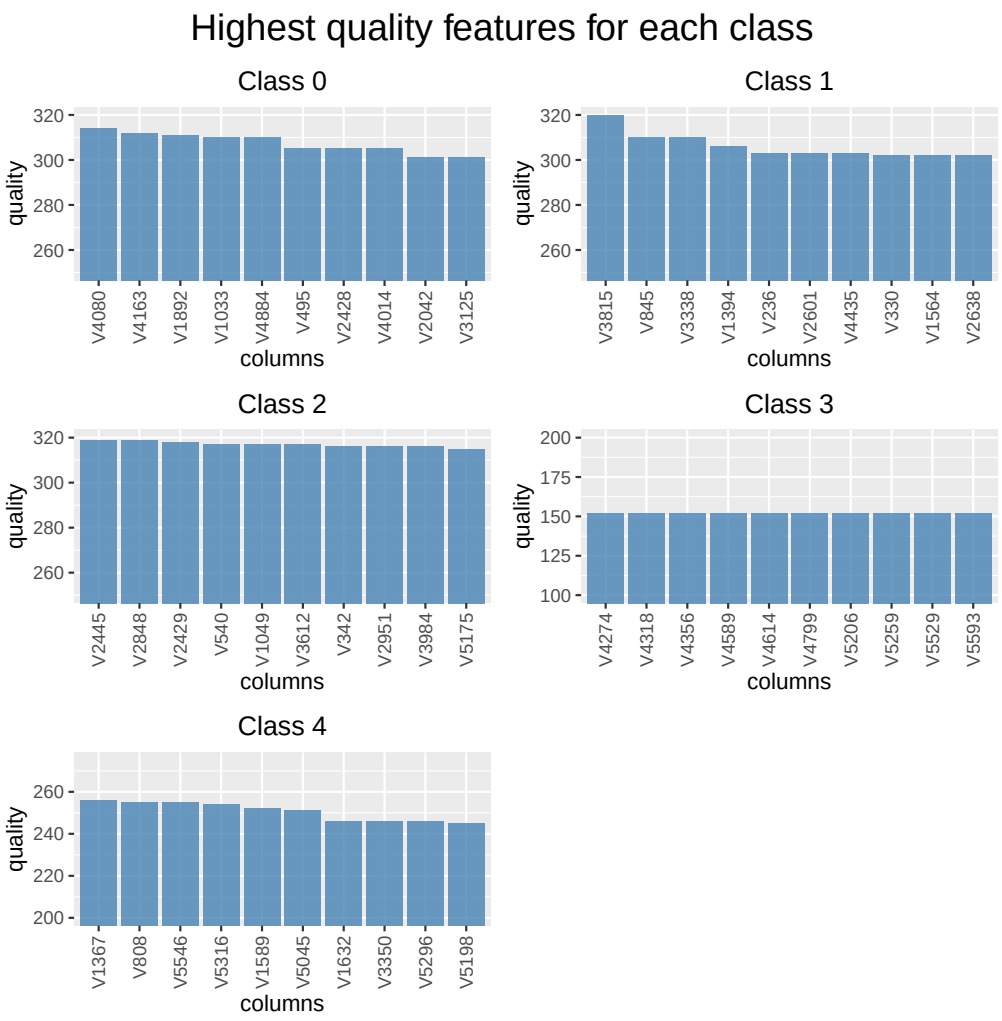

Figure 3.2. Highest quality score features for each class

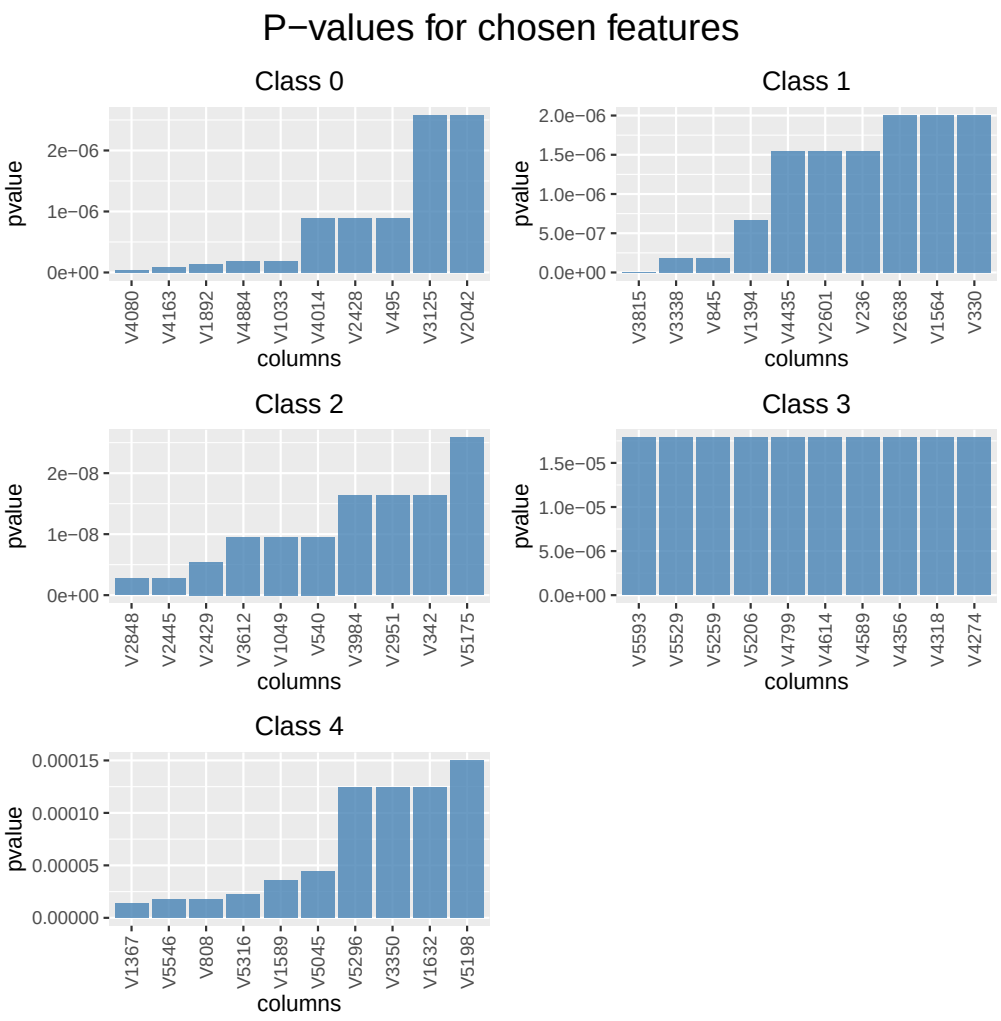

Figure 3.3. P-values for highest quality score features for each class

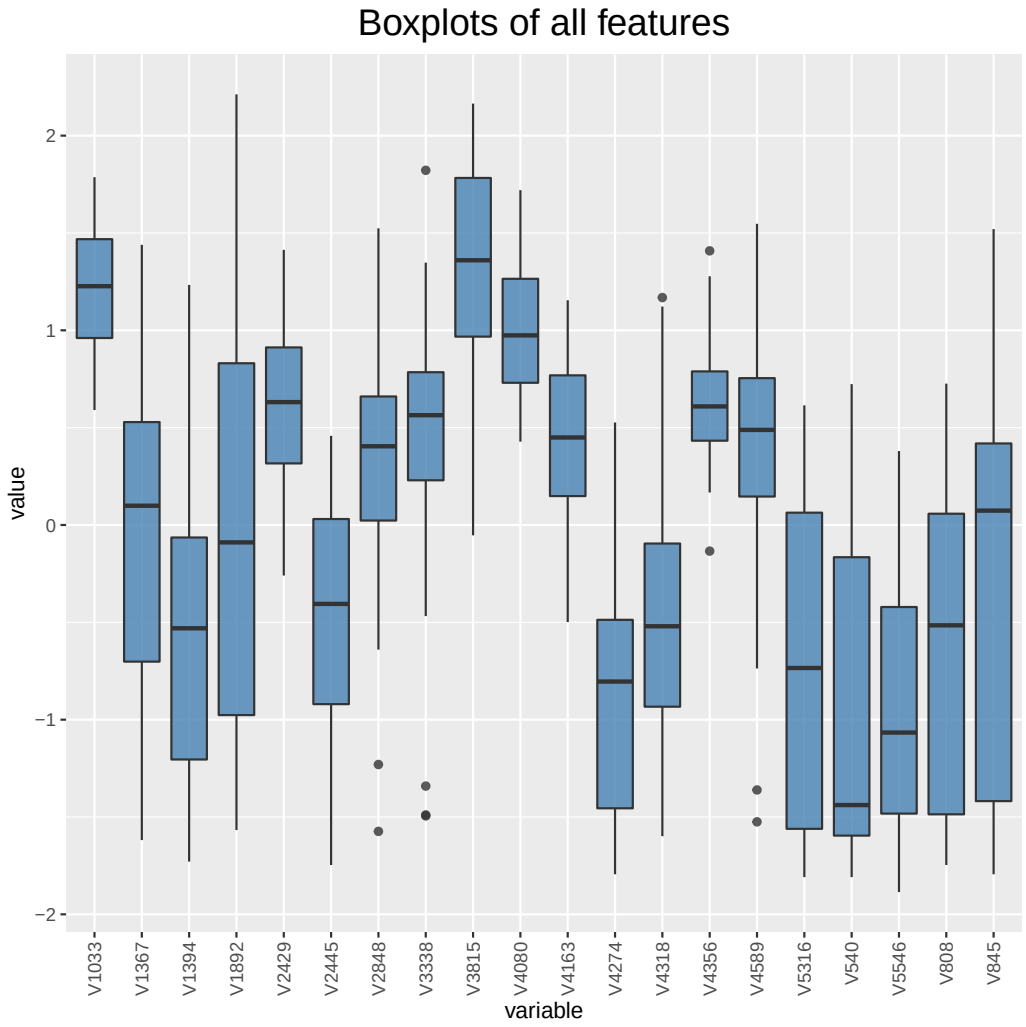

Figure 3.4. Boxplots of all features

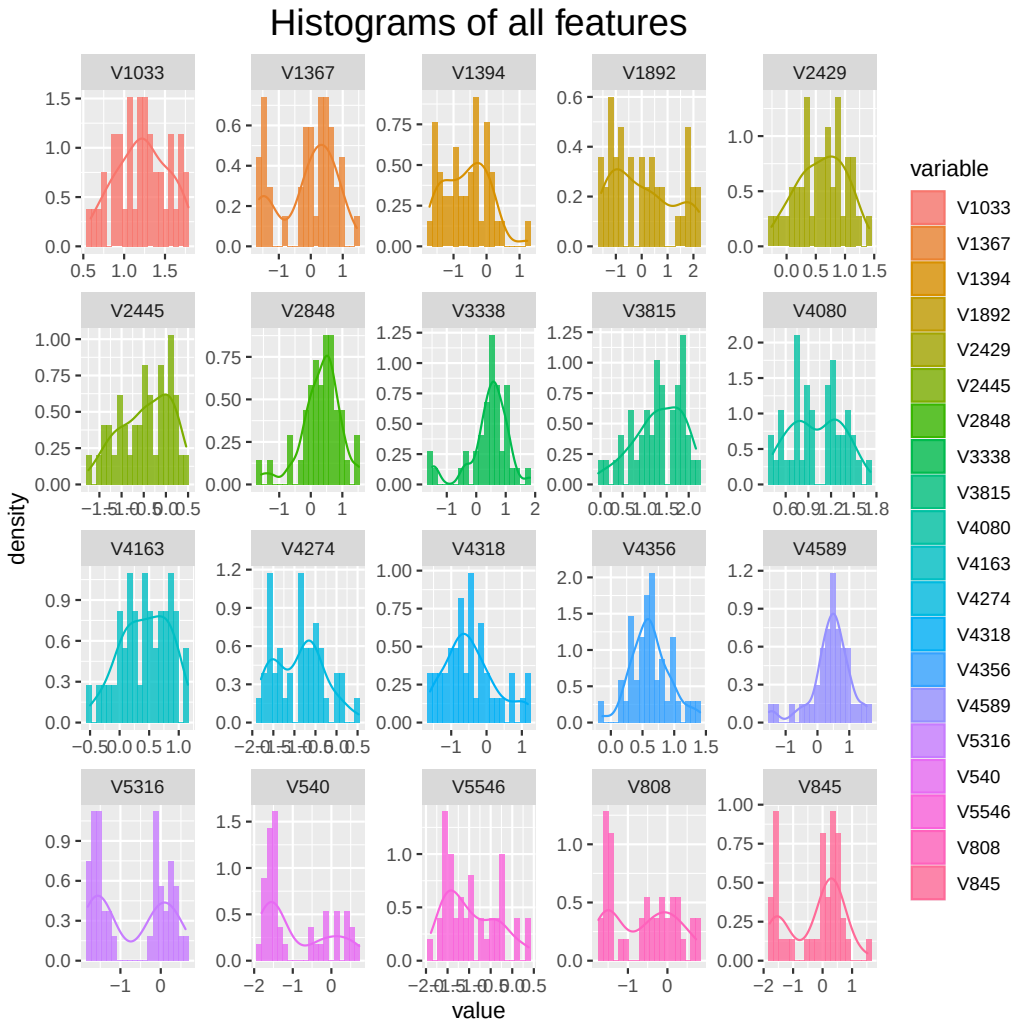

Figure 3.5. Histograms of all features

Column	p-value
V4589	3.2654277×10^{-6}
V4356	$4.2167381 \times 10^{-12}$

Table 3.2. P-values for selected features for testing normal distribution.

Column	p-value
V2429	0.0380464
V808	0.0307209
V4163	0.1550784
V4080	0.3916026

Table 3.3. P-values for selected features for testing uniform distribution.

we will first choose the 10 most important features for each class. Regarding the quality statistic values shown on the graph (3.2), different features have varying quality values for each class. Moreover, each class does not choose exactly the same genes as the most important ones. The same behavior is evident on the graph (3.3), where the same genes are presented for each class in terms of p-values. All p-values presented on the graphs are less than 0.05, allowing us to reject the null hypothesis that these features are not significant for specific classes.

Next, we will select the four most important features for each class and conduct the analysis on them. On the summary matrix below (3.1), we can see that most of the features contain negative values, while only features $V4080$ and $V1033$ do not. The ranges of feature values are similar - there is no gene with significantly bigger range. From the last column of the summary, we can see that features contain most observations in different value ranges.

In the boxplots of each feature (3.4), features $V2848$, $V3338$, $V4318$, $V4356$, and $V4589$ have outliers. We do not treat them as incorrect values, considering we only have 42 observations and 5 different types of tumors. Outliers might help us distinguish specific types of tumors. The boxplots of features are spread across different parts of the Y-axis - indicating a potential need for data standardization.

Reviewing the histograms of each feature (3.5), we can observe the distribution of each gene. Histograms of $V1033$, $V2429$, $V2445$, $V2848$, $V3815$, $V4163$, $V4318$, $V4356$, and $V4589$ features are unimodal, while histograms of $V1367$, $V1394$, $V1892$, $V4080$, $V4274$, $V5316$, $V540$, $V5546$, $V808$, $V845$, and $V3338$ features are multimodal. We suspect that the distributions of features $V4589$ and $V4356$ might be normal. Using the Kolmogorov-Smirnov test, we obtain a p-value less than 0.05 in both cases (table 3.2), indicating rejection of the null hypothesis about normal distribution. For features $V4163$, $V2429$, $V4080$, and $V808$, we will check if the distribution is uniform using the Kolmogorov-Smirnov test. For features $V2429$ and $V808$, the p-value is less than 0.05 (table 3.3), rejecting the null hypothesis about uniform distribution. However, for features $V4163$ and $V4080$, the p-value is greater than 0.05, so we cannot reject the hypothesis about uniform distribution of data.

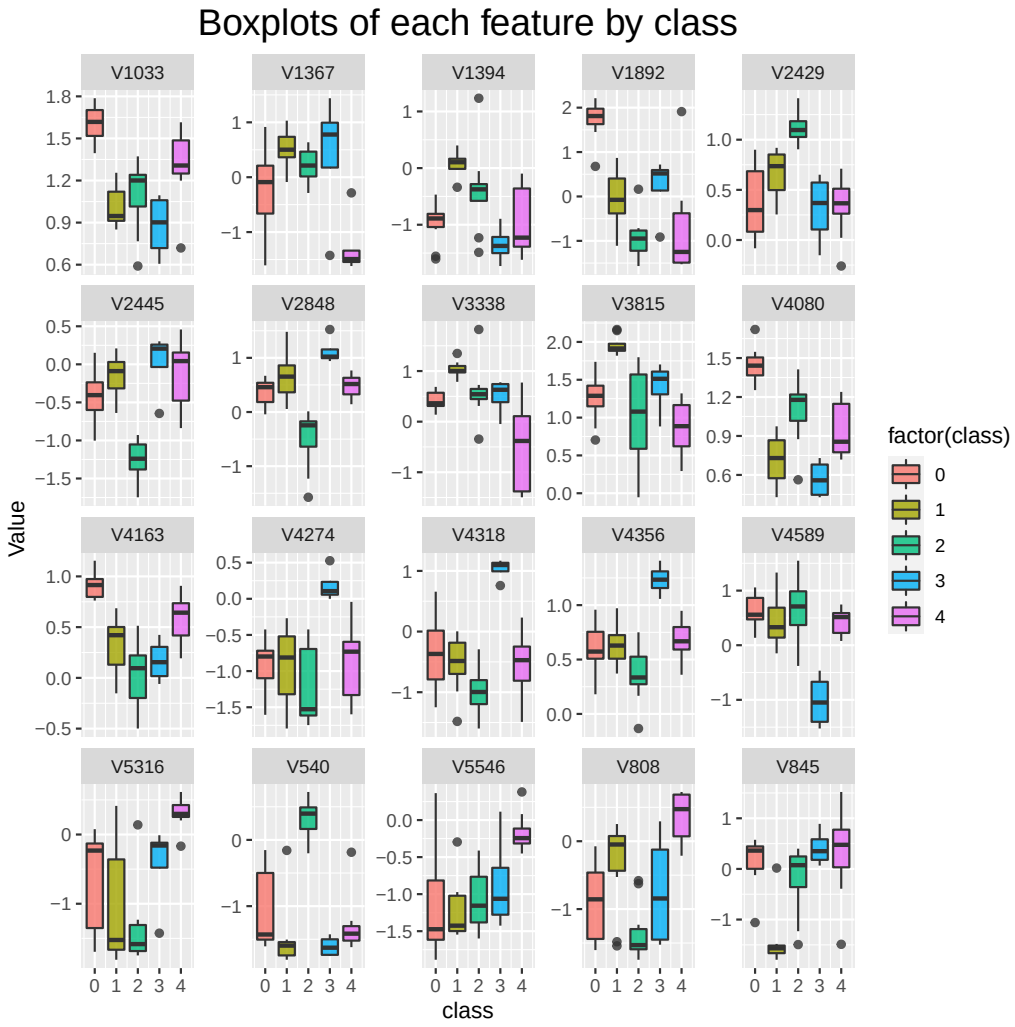

Figure 3.6. Boxplots of each feature grouped by class

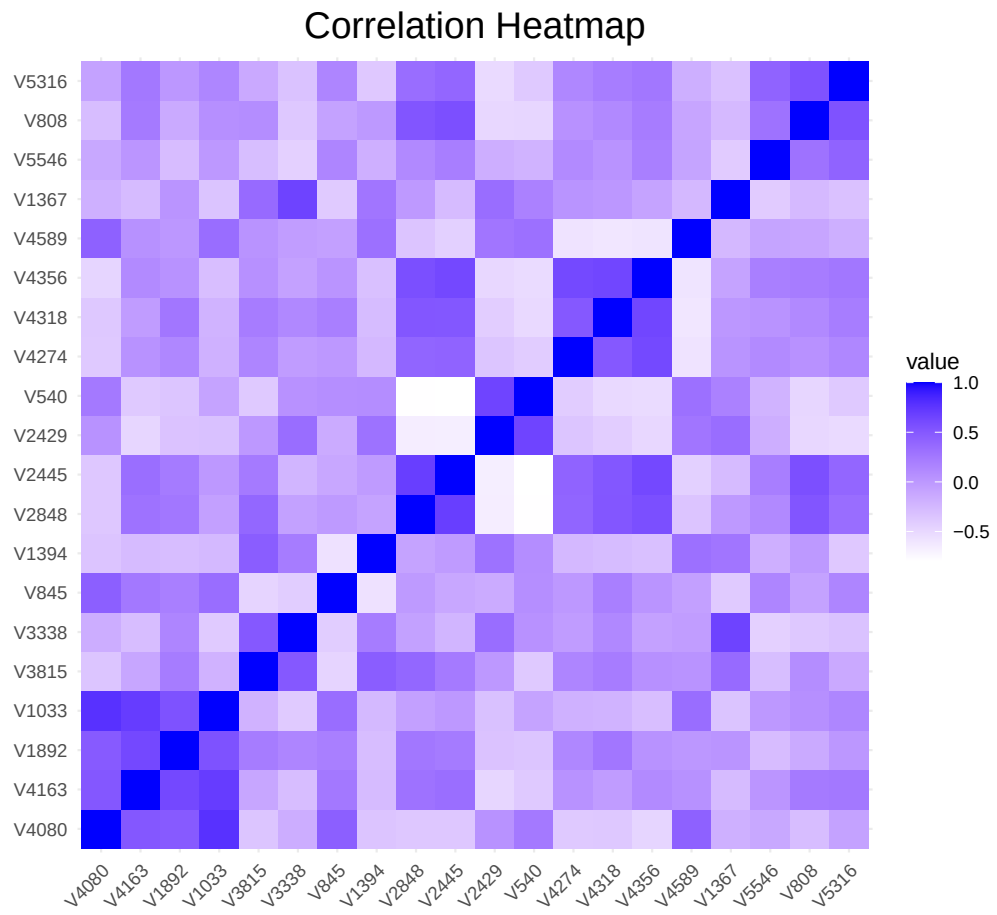

Figure 3.7. Correlation of selected features

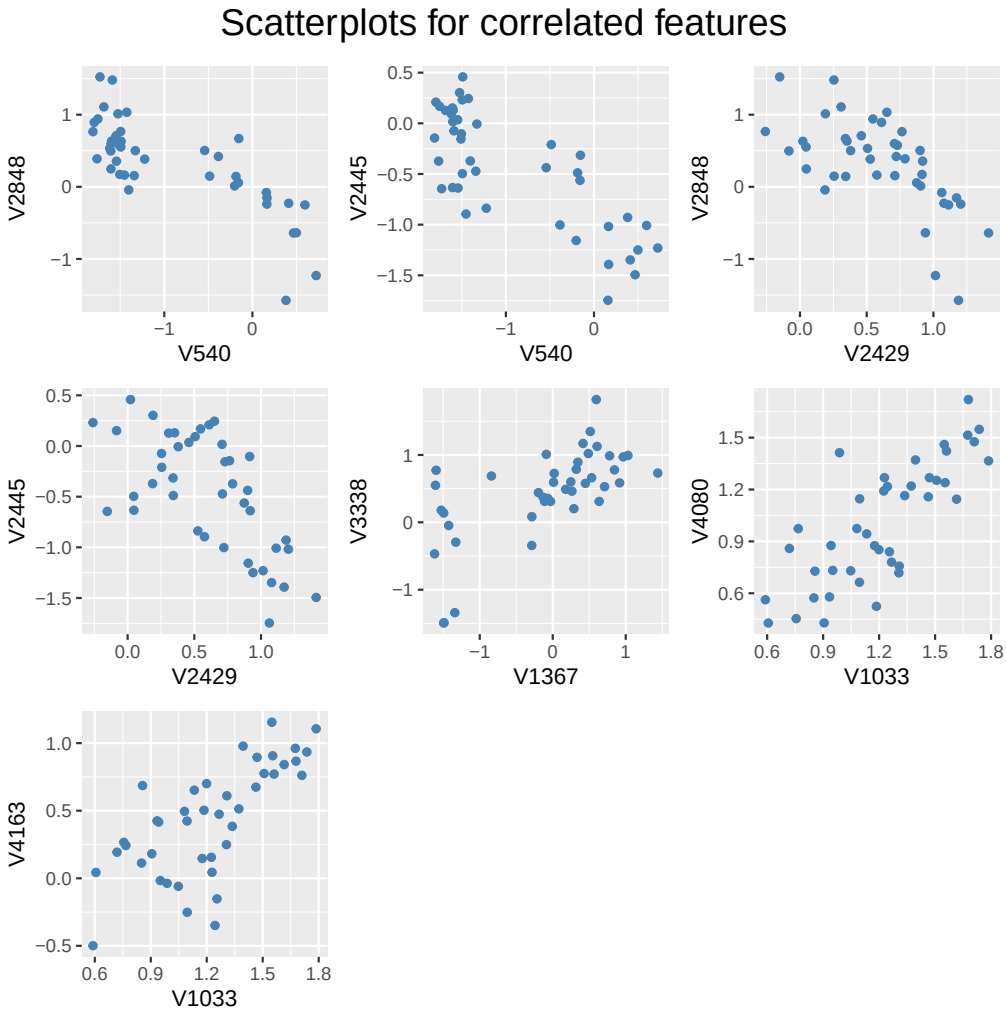

Figure 3.8. Scatterplots for correlated features

Now, we aim to perform an analysis within classes to investigate the discriminative ability of our features and their importance. On the graphs depicting boxplots of each feature concerning class labels (3.6), several important conclusions can be drawn.

1. Feature $V1033$ is discriminative for classes 0 and 4.

2. Feature $V1394$ is discriminative for class 1, feature $V1892$ for class 0, feature $V2429$ for class 2 and feature $V2445$ for class 2.

3. Feature $V2848$ is discriminative for class 2 and 3, feature $V3338$ for class 1 and 4, feature $V3815$ for class 1.

4. Feature $V4080$ is discriminative for class 0, $V4163$ for class 0, $V4274$ for class 3 and $V4318$ for class 3.

5. Feature $V4356$ is discriminative for class 3, $V4589$ for class 3, $V5316$ for class 4 and $V540$ for class 2.

6. Feature $V5546$ is discriminative for class 4, $V808$ for class 4 (to a lesser extent) and $V845$ for class 1.

From the conclusions drawn from the mentioned boxplots, it is evident that each feature is discriminative for at least one class. This underscores the importance of the chosen genes by our feature selection method and its utility.

We would also like to investigate the correlation between our features. On the correlation heatmap (3.7), some features show strong correlations with each other. For the strongest correlations, we can examine the scatterplots (3.8).

1. For pairs of features $V3338$, 1367 and $V4080$, 1033 and $V1033$, 4163, we observe positive correlation.

2. for pairs of features $V2848$ and 540, 2445 and 540, 2848 and 2429, 2445 and 2429, we observe negative correlation.

4. Classification

We will face problem of multiclass classification. We will investigate following models:
— K Nearest Neighbors,
— Linear and Quadratic Discriminant Analysis,
— Multinomial Logistic Regression,
— Decision Tree,
— Random Forest.

We split the data into learning (2/3) and test (1/3) sets once for the classification part. Taking into account the class imbalance problem we will use `createDataPartition` function from the `caret` package where stratify is included.

We will perform leave-one-out cross-validation in order to find the best parameters of chosen models. In each step of cross-validation we will perform feature selection procedure on the training subset of the learning set. We will choose 40 best features with respect to each class - 200 features in total. Thereupon we will avoid information leakage from the validation subset. Therefore, will treat feature selection process as an integral part of the model pipeline. Similarly, the final models will perform feature selection only on the given learning set. Additionally, we incorporate data standardization also being careful about information leakage from the validation and test sets.

We will also investigate the significance of choosing best features. We will check if randomly chosen features will impact the model performance.

After choosing the best models we will compare their missclassification errors and stability.

4.1. K Nearest Neighbors

The first model we will consider is the K Nearest Neighbors algorithm. We will try to find the optimal value of the k parameter. In order to do that will compare the average missclassification errors based on the leave-one-out cross-validation. We will investigate up to 10 nearest neighbors.

In the graph 4.1 we can see that the more neighbors we consider the greater the error gets. It applies to both train and validation errors. We choose $k = 1$ as the optimal value of the parameter because in such case we get the smallest value of the misclassification error on the validation set. It is worth noticing that for $k = 1$ we get the smallest error on the train set but since we also have the smallest error on the validation set we will not consider this case as an overfitting.

We will now consider similar procedure but without feature selection. We will train our model on 200 randomly chosen features. Based on the graph

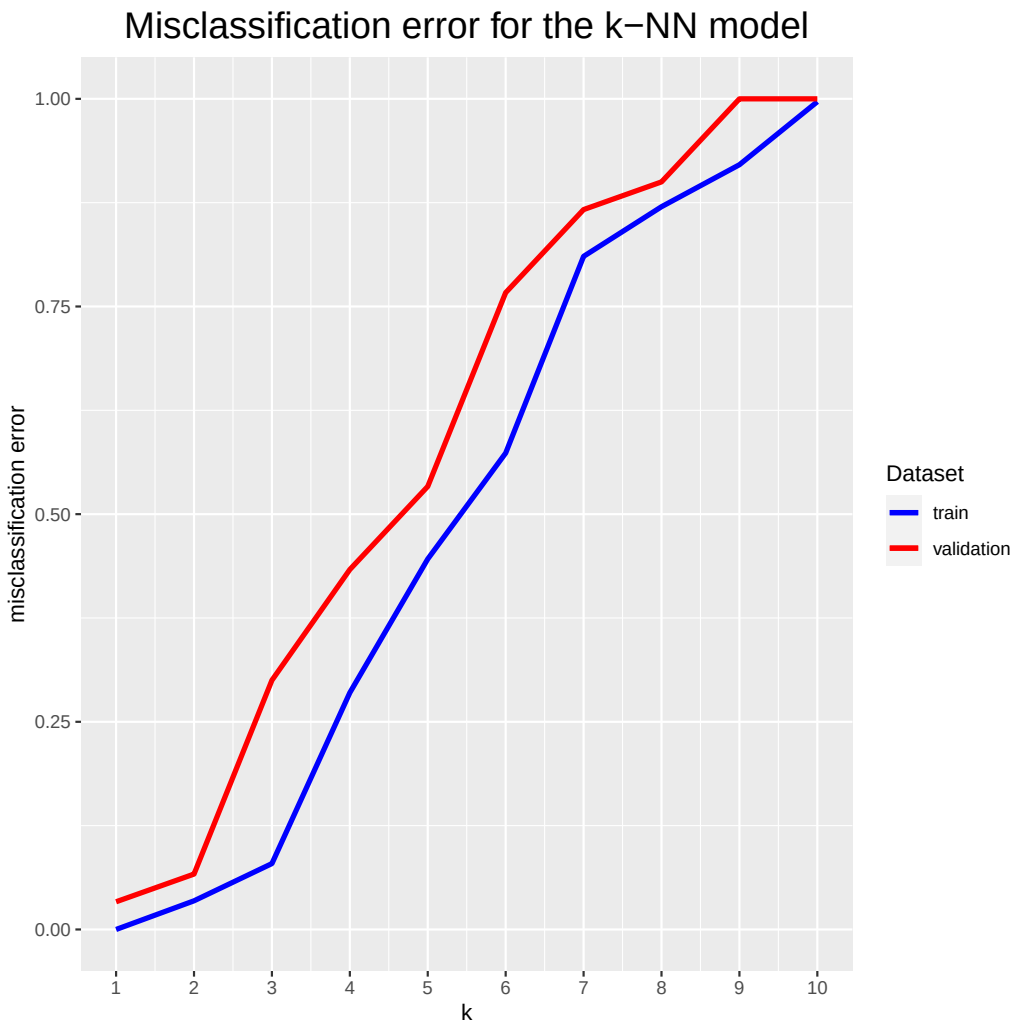

Figure 4.1. Misclassification errors for train set and validation set for kNN model

?? we can yield the same conclusion about the choice of parameter k, the optimal value is $k = 1$. However, it seems that we get greater misclassifcation error in comparison to the model with feature selection.

```
## Error in seq.default(0, max(log2(errors_rf$ntree)), by = 1):
nie znaleziono obiektu 'errors_rf'
## Error in eval(expr, envir, enclos):  nie znaleziono obiektu
'errors_knn_plot_random'
```

4.2. Linear Discriminant Analysis and Quadratic Discriminant Analysis

Next, we will discuss two linear classification methods - LDA and QDA. Both of them have assumption about multivariate normality of the data. In the case of LDA we assume that covariance matrices for all classes are the same, and for QDA we estimate them for each class separately. However, our

data does not have many observations, one class has only 4 records - it is not correct to make estimations of the covariance matrix on such small sample. Additionally, the number of predictor variables is much greater than the size of the smallest group in our data, therefore it may not be the best idea to use linear discriminant analysis in our case[2].

4.3. Multinomial Logistic Regression

Instead of LDA and QDA we will use another linear model - Multinomial Logistic Regression, which imposes fewer assumptions on data. In case of this model we will not search for optimal parameters. Contrary to other models, in this case we will pass less features to the model - 10 for each class, 50 in total due to model limitations.

4.4. Decision Tree

We will now consider another type of classification algorithm - Decision Tree with minsplit = 1, minbucket = 1, which is reasonable for our data. One of possible parameters of the Decision Tree model is the complexity parameter *cp*, which is responsible for the size of the tree. It corresponds with tree pruning in terms of compromising between classification accuracy and model complexity.

We will consider *cp* values from 0.02 to 0.2. In the plot of misclassification errors (4.2) we can see that on the validation set we get the smallest errors for *cp* values lower than 0.08. We observe that we do not have the case of overfitting because when we have the smallest values of train errors, we also have the smallest values of validation errors. For our final model we will choose the complexity parameter equal to 0.02.

In the plot (4.3) for the case of random subset of features we can observe that the model performed the best for the lowest values of *cp*. However, the values of misclassification errors, especially for the validation set, are greater that the ones for the model with feature selection.

4.5. Random Forest

In terms of the Random Forest we want to investigate the influence of the m parameter - number of features selected randomly sampled as candidates at each split. Will consider the forest of 10 trees. We will see how reduction of number of the most important features included in each tree impacts the performance of the model. Since in feature selection we consider 200 best features, we will examine the range of the m parameter from 25 to 200. From the plot of misclassification errors (4.4) we can see that we obtain the best model for $m = 175$. However, we can notice that the choice of the m parameter does not destabilize the model strongly.

Figure 4.2. Misclassification errors for train set and validation set for the Decision Tree

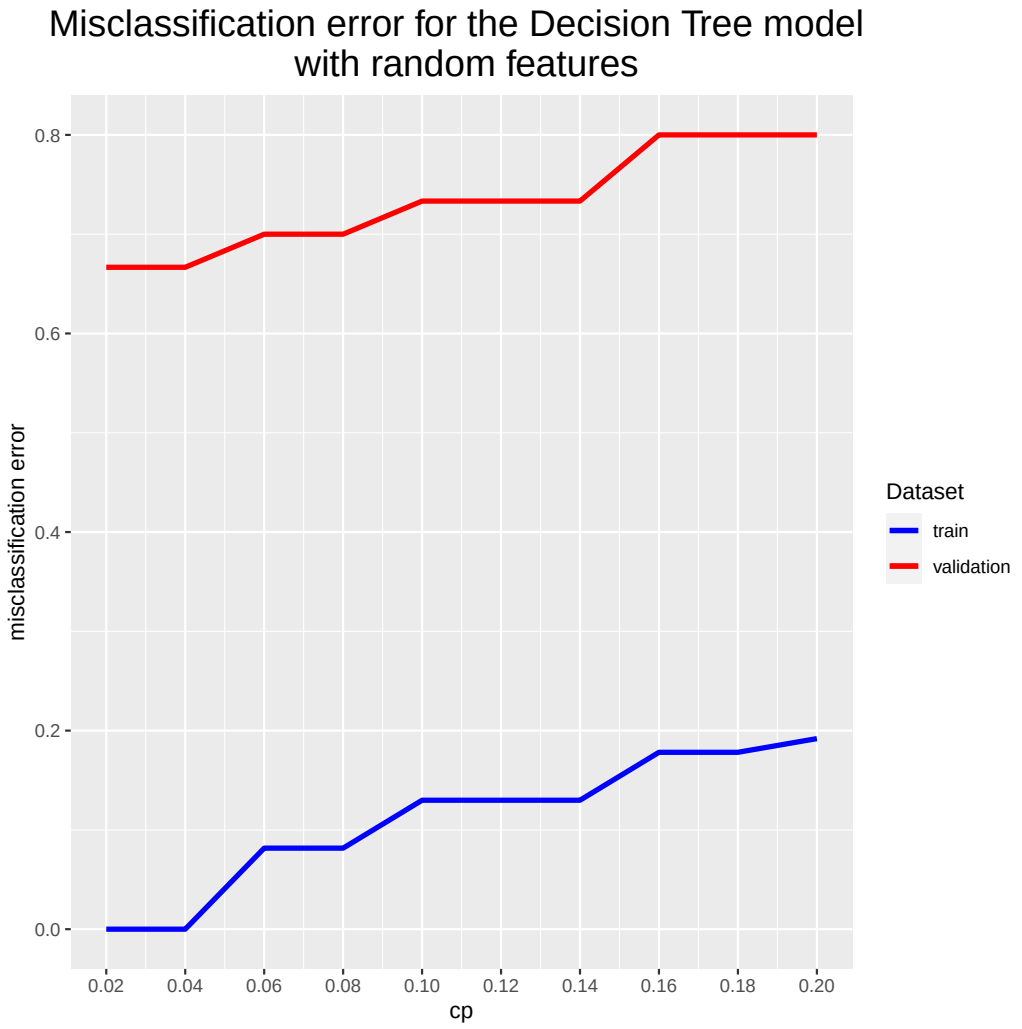

Figure 4.3. Misclassification errors for train set and validation set for the Decision Tree with random features

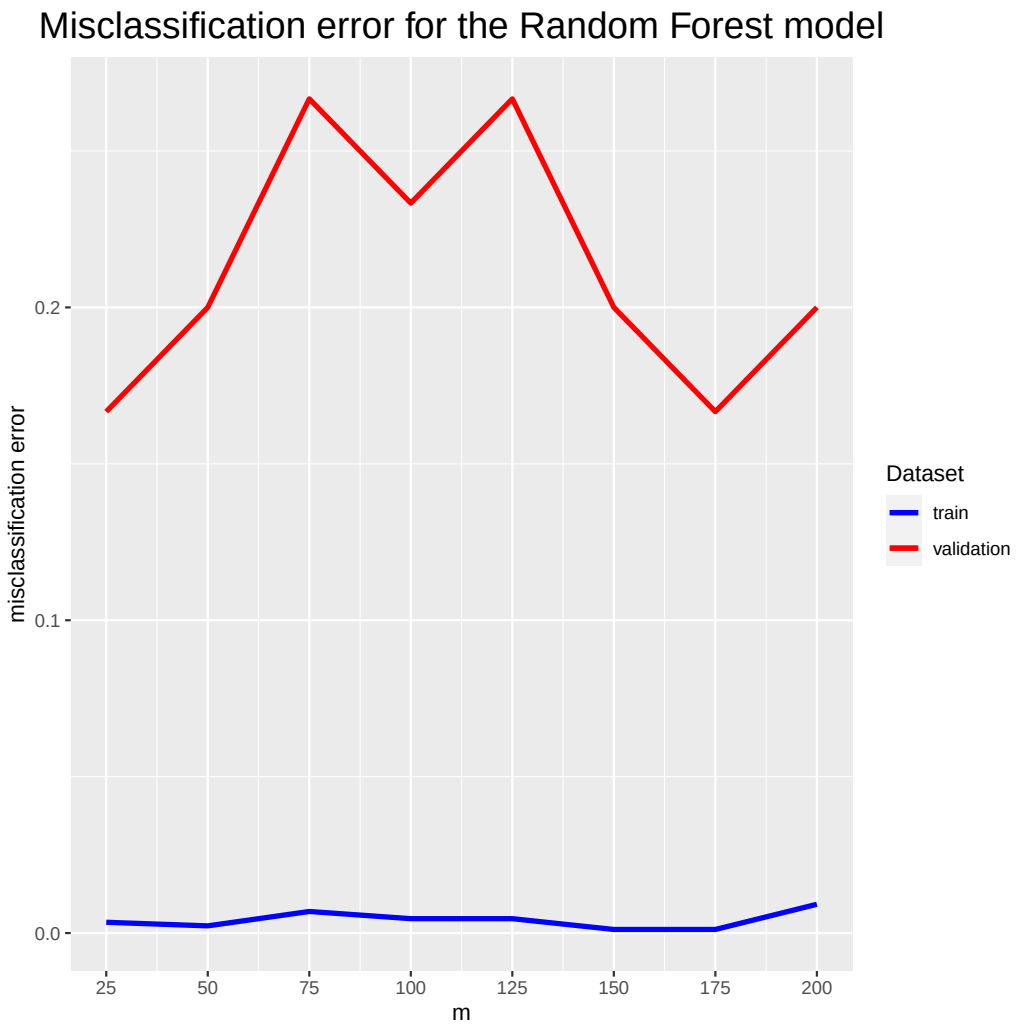

Figure 4.4. Misclassification errors for train set and validation set for the Random Forest

In the graph (4.5) for random features we see that we get the lowest validation error for maximum value of the m parameter, $m = 200$. Again, the model with random features seems to perform worse.

4.6. Comparison of the models

In previous sections we have chosen the models with their parameters. Now we will train them on previously generated learning set and evaluate them on the test set. For each model we will compare the performance with feature selection and without. From the table (4.1) we can read that there are no differences between models in terms of fitting the learning set - all errors are equal to 0. However, the most important thing for us is the generalization ability and performance on the test set. Comparing the misclassification errors on the test set (4.2) we can see that all models with feature selection, with Multinomial Logistic Regression in the lead, performed quite well in our

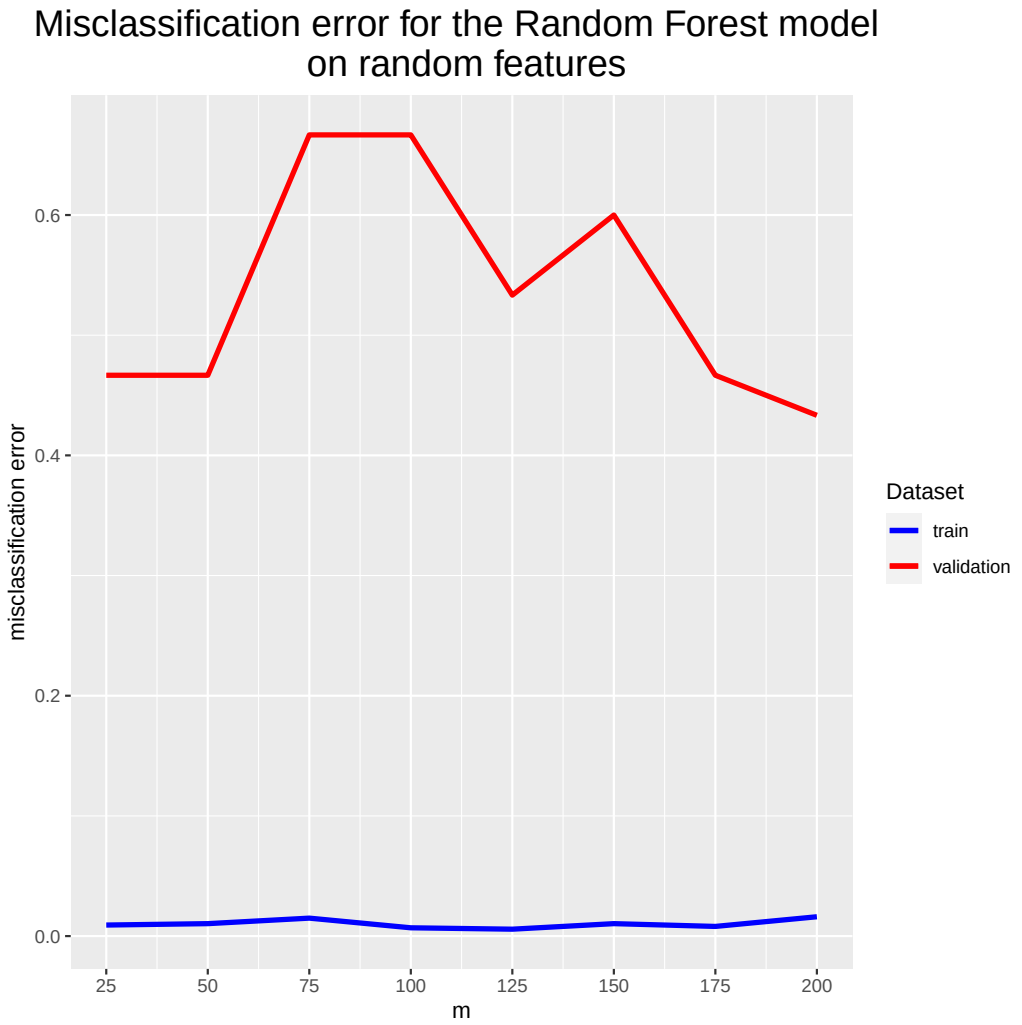

Figure 4.5. Misclassification errors for train set and validation set for the Random Forest on random features

model	feature selection	no feature selection
K Nearest Neighbors	0	0
Multinomial Logistic Regression	0	0
Decision Tree	0	0
Random Forest	0	0

Table 4.1. Misclassification errors on the train set for different classification models

model	feature selection	no feature selection
K Nearest Neighbors	0.1666667	0.1666667
Multinomial Logistic Regression	0.0833333	0.0833333
Decision Tree	0.25	0.5833333
Random Forest	0.1666667	0.4166667

Table 4.2. Misclassification errors on the test set for different classification models.

case. We can also see positive impact of future selection procedure for the Decision Tree and Random Forest.

We would like to address the class imbalance problem that we had in our data. Since we have multiclass classification problem, we will investigate the macro-averaged F1 score. It is computed by taking the arithmetic mean of all the per-class F1 scores. This method treats all classes equally regardless of their support values. Error in classification in each class has the same negative impact. The model with the highest score is the Multinomial Logistic Regression with feature selection and the lowest is for the Decision Tree without feature selection (4.3). The Multinomial Logistic Regression model not only got the lowest error on given test set but also performed the best in terms of dealing with the imbalance class problem. Overall, all models with feature selection got quite good results. We got one NA value which may be caused e.g. by lack of predictions for one class.

We are also interested in stability of chosen models. We will iteratively generate 100 learning and test subsets and perform model fitting. The misclassification errors are visualized on the boxplots (4.6, 4.7). We can see that all models but Random Forest fitted the train sets ideally, resulting in errors equal to 0. We can conclude that the k-NN algorithm seems to have the lowest mean error. On the other hand, the one that performs the worst is the Decision Tree. The algorithm with the lowest dispersion and therefore the most stable is the k-NN. The performance of the Multinomial Logistic Regression and Random Forest is approximately the same in terms of stability.

model	feature selection	no feature selection
K Nearest Neighbors	0.7933333	0.8333333
Multinomial Logistic Regression	0.9111111	NA
Decision Tree	0.6266667	0.4833333
Random Forest	0.8111111	0.6333333

Table 4.3. Macro-averaged F1 score on the train set.

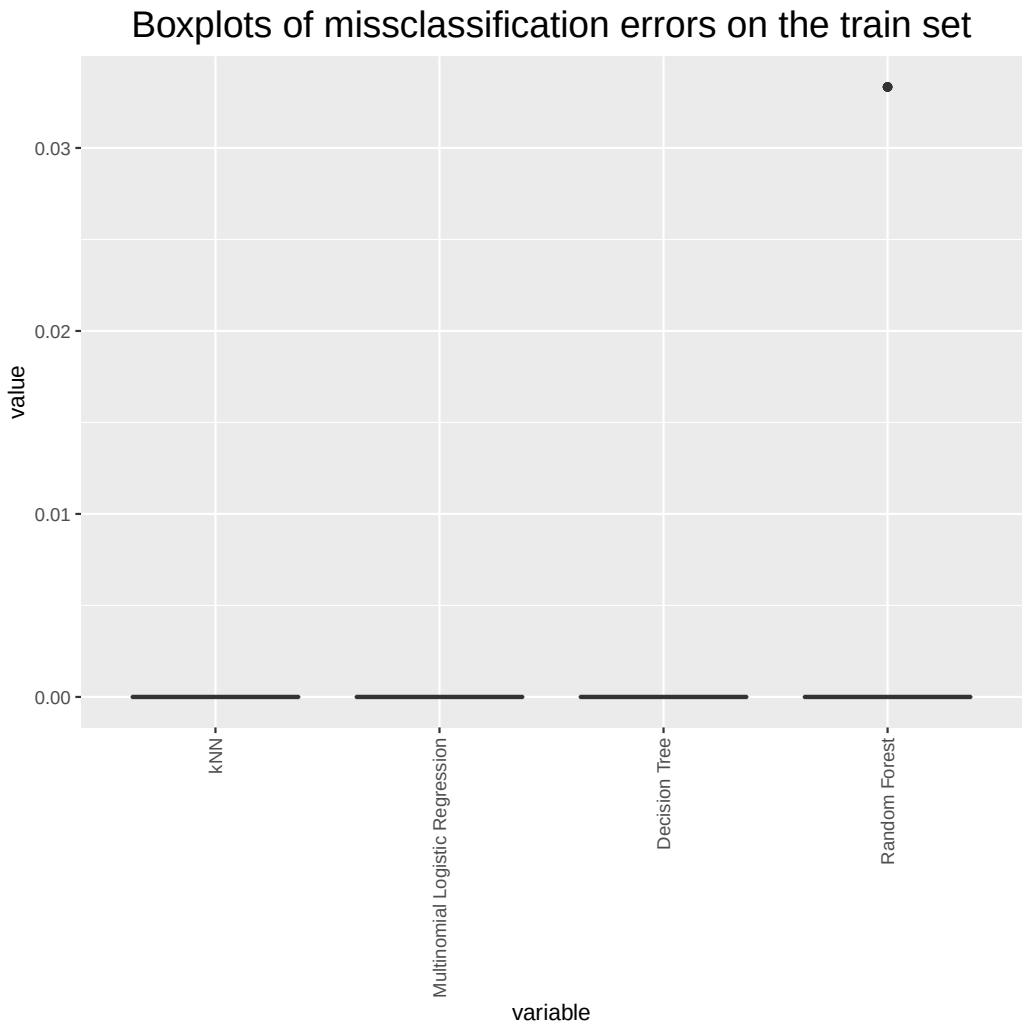

Figure 4.6. Boxplots of misclassification errors on the train set for different models

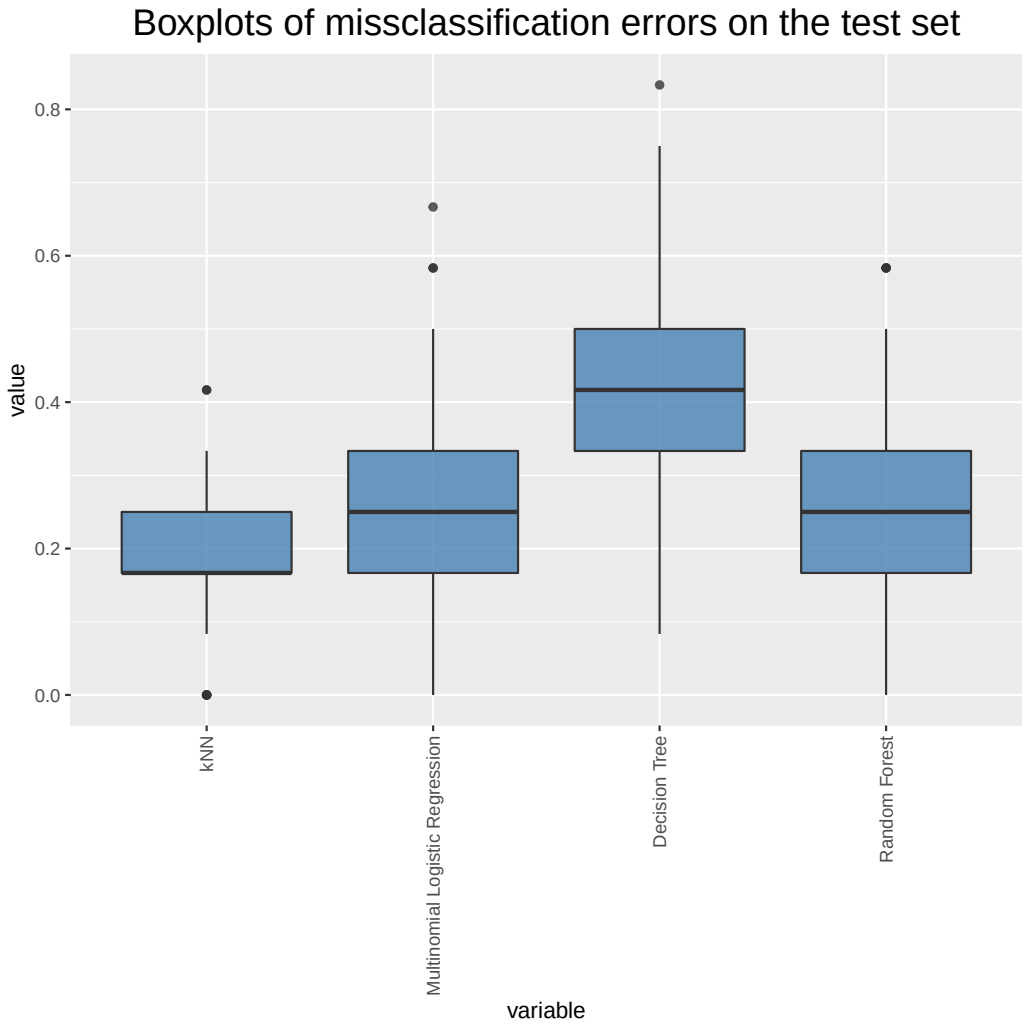

Figure 4.7. Boxplots of misclassification errors on the test set for different models

5. Stability of feature selection method

We would like to investigate the stability of chosen feature selection model described in section 2. We can do it by looking at the graph 5.1 based on selecting 40 best features for each class via leave-one-out cross-validation on the learning set. We can observe that some features have been selected significantly more often then others. What is more, the majority of features has been selected only few times. We can conclude that the algorithm is not stable in our case.

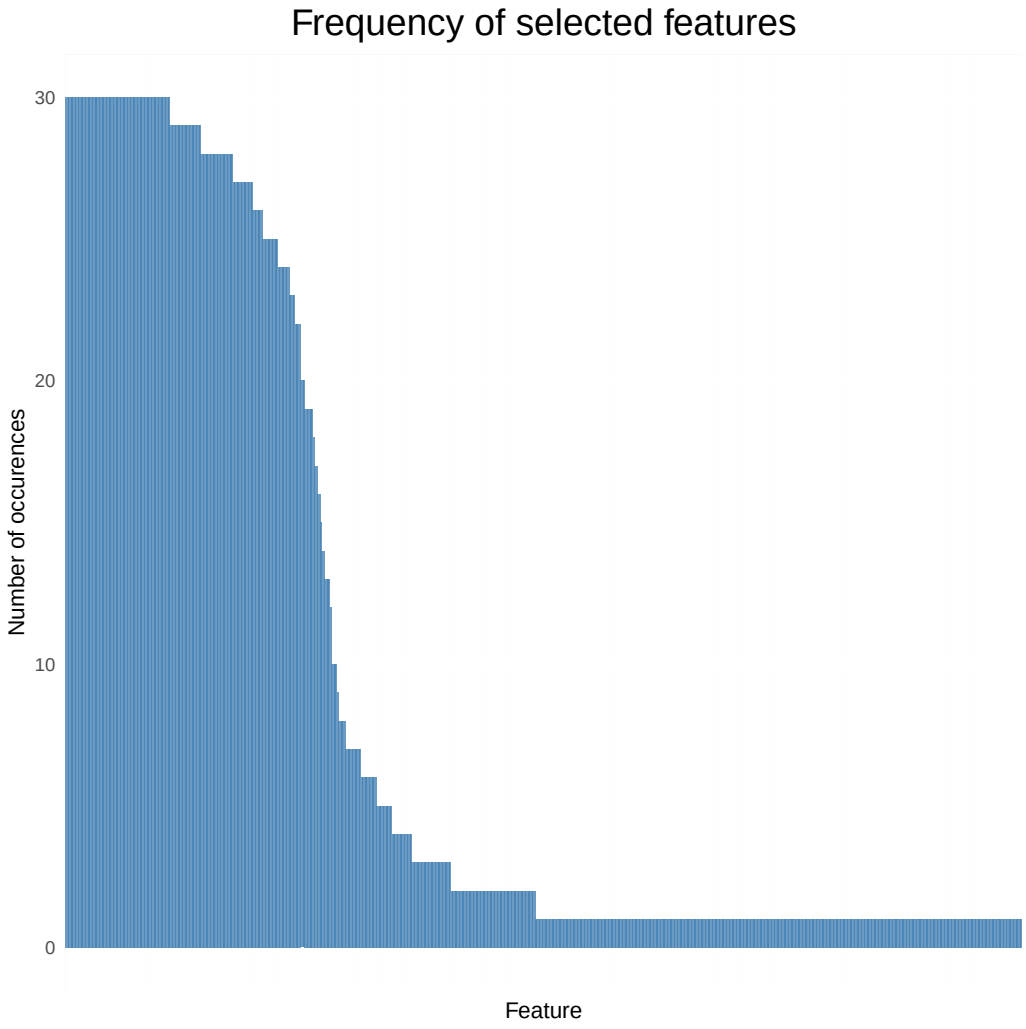

Figure 5.1. Frequency of choosing each feature

6. Conclusions

We conclude that the feature selection method is not stable. However, we observe that in general models with feature selection tend to perform better that the ones without, which indicates its usefulness.

From exploratory data analysis on the 20 most important features we were able to investigate the distribution of each of them, formulate assumptions about their normality or uniformity and verify those assumptions with help of statistical tests. We also checked potential discriminative ability of selected features.

Even though our data has $p \gg n$ problem and class imbalance problem we were able to construct models that gave us satisfying results in terms of the misclassification error and macro-averaged F1 score. Overall, the method that performed the best in our case in terms of accuracy and stability is the K Nearest Neighbors algorithm with $k = 1$ and features selected with help of the proposed feature selection procedure. The Random Forest with $m = 175$ and Multinomial Logistic Regression gave us slightly bigger errors, but still we consider them as better algorithms than the Decision Tree with $cp = 0.02$, minsplit = 1, minbucket = 1. We rejected LDA and QDA algorithms. After analysis we conclude that these are not appropriate approaches for our data.

7. Further research suggestions

In the realm of microarray data analysis and classification, exploring new methods for selecting a few marker components of the gene subset that determine the type of tumor is worth investigating. Moreover, the feature preselection algorithm employed in the project can be evaluated not only in comparison to models with randomly chosen features but also to models trained on all 5597 features. While computationally demanding, this approach will yield explicit results regarding the usefulness of the method.

Furthermore, it might be beneficial to consider additional algorithms such as SVM, neural networks, etc. The algorithms already used in the project can be further investigated, employing leave-one-out cross-validation while exploring a grid of more than one parameter.

Addressing the class imbalance problem, specific procedures like oversampling can be applied and examined.

In medical diagnostics, the classification of tumor types can be widely used to provide an additional perspective to professionals. The models developed and the analysis conducted can be instrumental in further research, offering new data-driven perspectives on today's medicine.

8. Bibliography

1 Marcel Dettling and Peter Buhlmann, Boosting for tumor classification with gene expression data, Seminar fur Statistik, ETH Zurich, CH-8092, Switzerland.

2 Büyüköztürk, Çokluk-Bökeoğlu (2008). Discriminant function analysis: Concept and application. Egitim Arastirmalari - Eurasian Journal of Educational Research, 33, 73-92.