University of Groningen

Introduction to Data Science

Classification of acute myeloid leukemia patients

Group 16:
Otte Tjepkema *(s3237184)*
José Rodrigues *(s4169328)*
Andrei Miculita *(s4161947)*
Robert Riesebos *(s3220672)*

October 3, 2019

Contents

1 Introduction

Flow cytometry is a laboratory method that is used to identify and analyse particular components within cells. In flow cytometry a sample of cells suspended in a fluid is guided through a laser beam. Depending on the characteristics of the cell the light is scattered in different ways, which is picked up by a photodetector. Often fluorescent dyes are added to this process to easier distinguish the signals. This method has the ability to analyse the cells at a high rate of speed, up to thousands of cells per second [1]. Therefore it is well suited to detect cells with cancer related surface markers from blood samples. The data obtained from flow cytometry is often complex and high dimensional and therefore computer assisted decision systems are needed for diagnostic purposes [2].

In this report we will use a flow cytometry dataset to distinguish acute myeloid leukemia (AML) patients from healthy patients. A total number of 359 patients are featured in this dataset of which 43 have AML. The first 179 patients have been labeled and will be used to train various classifiers. The other 180 subjects still need to be classified. We will first perform an exploratory analysis to see the effects of different kinds of preprocessing on the data. After that we will use two classification methods, KNN and DT, using the different preprocessing methods to find the best configuration for the classification of the unlabeled patients. Finally, we will train a heterogeneous ensemble of classifiers.

2 Question 1.1 Descriptive and Exploratory Analysis

2.1 Sub-question 1.1(a) Investigate the features

We first performed a principal component analysis (PCA) on the dataset. Since the features in the dataset have widely varying scales, we have scaled all of the data to unit variance during the PCA. A scree plot of the results can be seen in figure 1. We see that the first five principal components explain 61% of the variance in the data. To explain 90% and 95% of the variance in the data we would need 25 and 38 principal components respectively.

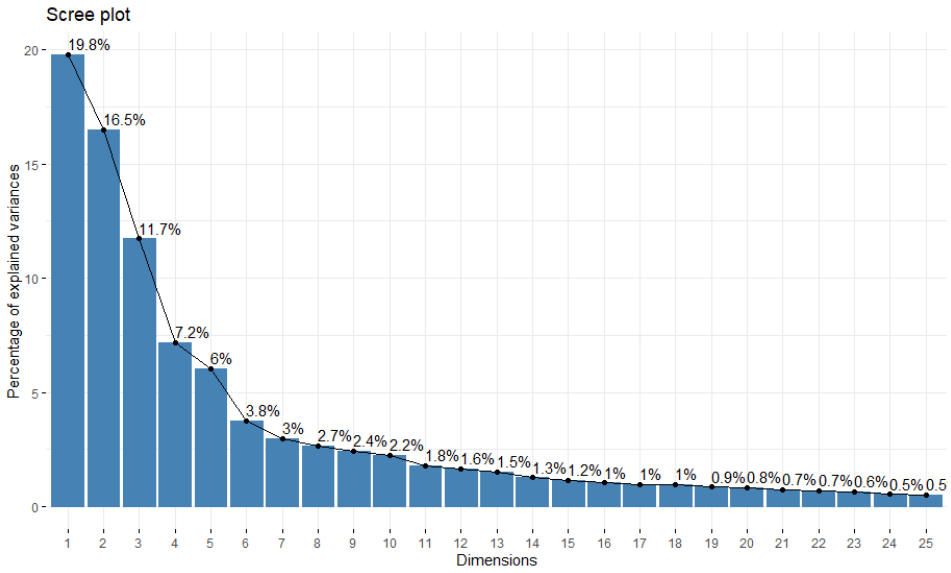

Figure 1: Scree plot of PCA components of the training set

To view which features are most important we have performed an ANOVA analysis. The most important features will have high F values, since these have the most variance between the two groups, which is important for classification. A table listing the best and worst features with the corresponding F values can be found in table 1.

Best features	F value	Worst features	F value
V134	190.5	V77	3.762e-02
V11	160.7	V18	4.886e-03
V133	159.6	V110	3.509e-03
V123	156.3	V73	1.838e-03
V122	137.1	V16	9.939e-04

Table 1: Table showing the F values for the five best and worst features from the ANOVA analysis.

We see that there is a large difference between the F values of the highest ranking and lowest ranking features. Therefore it would be a wise choice to eliminate the features with low F values before performing classification. This will reduce the dimensionality of the dataset, which will shorten computation time and improve the performance of the classifiers. Boxplots of the three best and worst features can also be seen in figures 2-7, where we can clearly see that the variance between groups is much larger for the best features, compared to the worst features.

2.1.1 Boxplots of best features

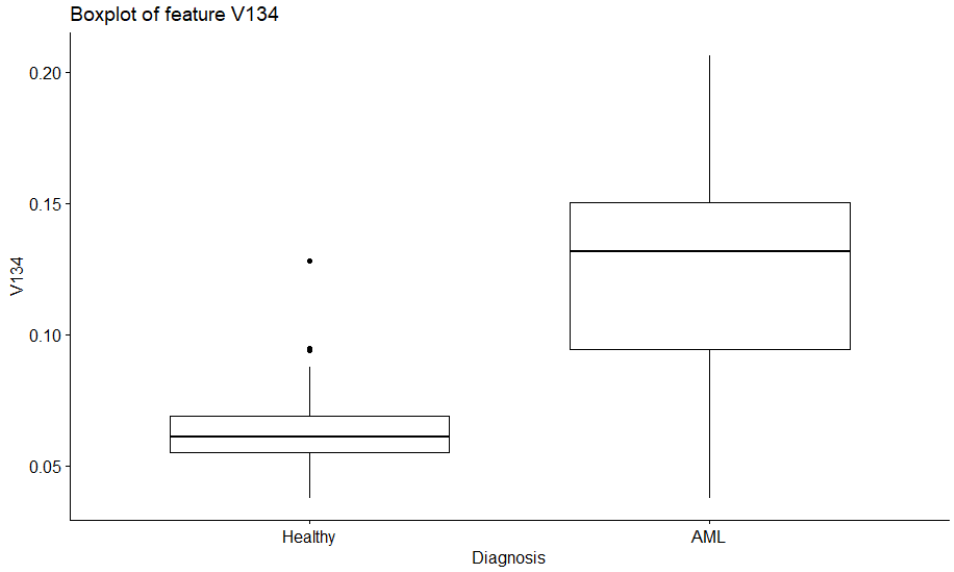

Figure 2: Boxplot of feature V134

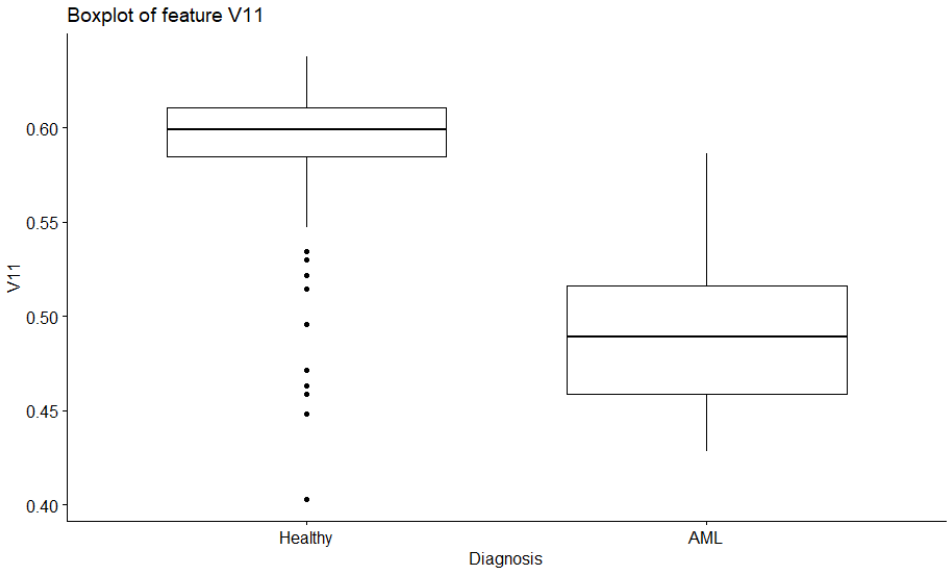

Figure 3: Boxplot of feature V11

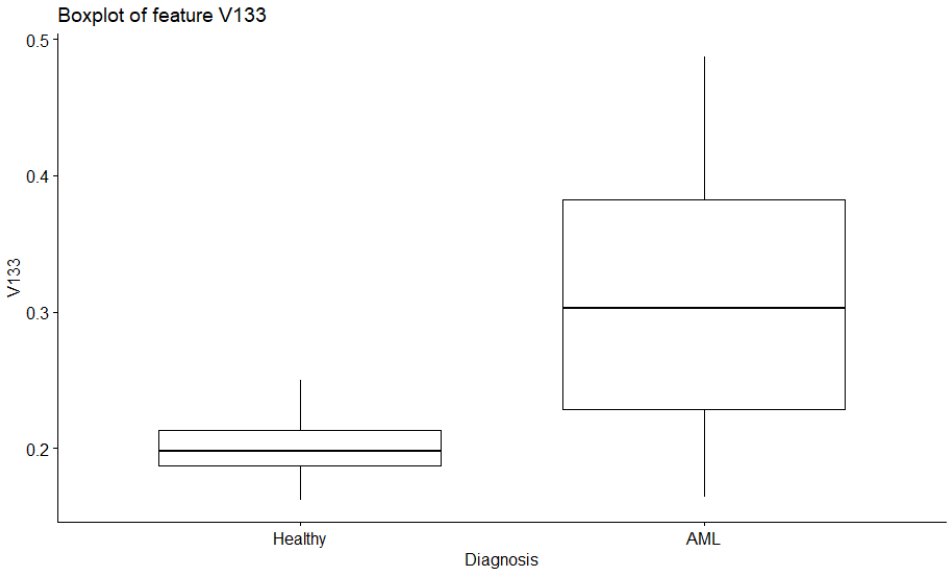

Figure 4: Boxplot of feature V133

2.1.2 Boxplots of worst features

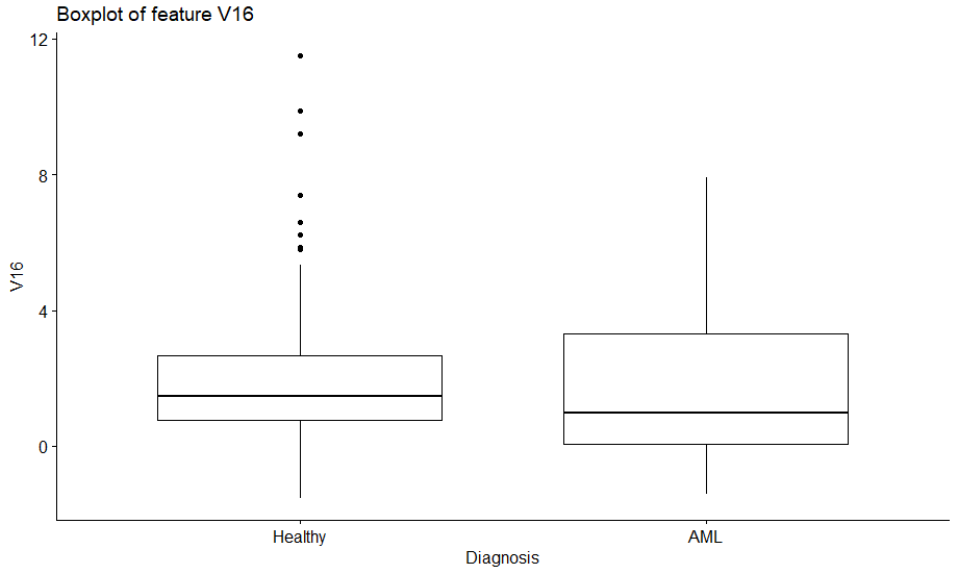

Figure 5: Boxplot of feature V16

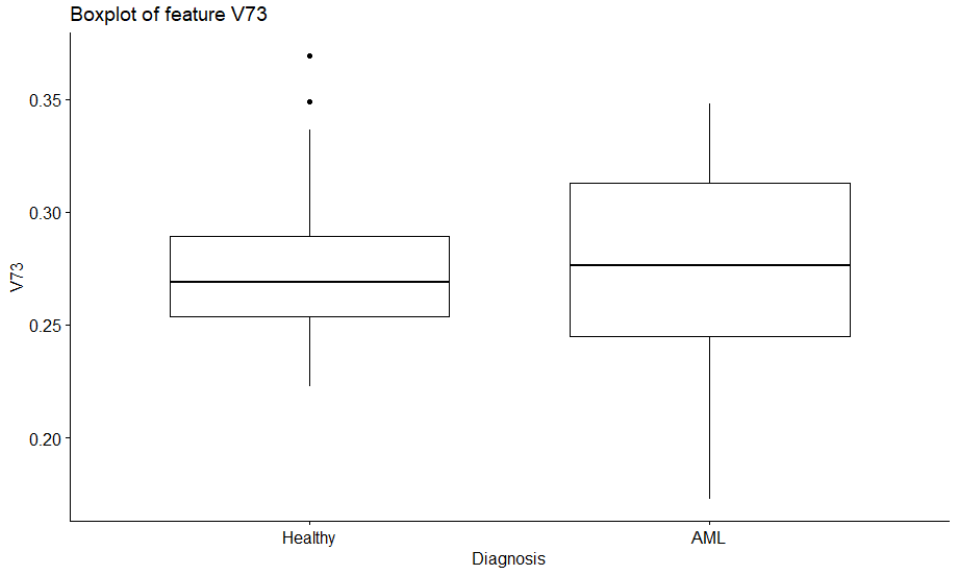

Figure 6: Boxplot of feature V73

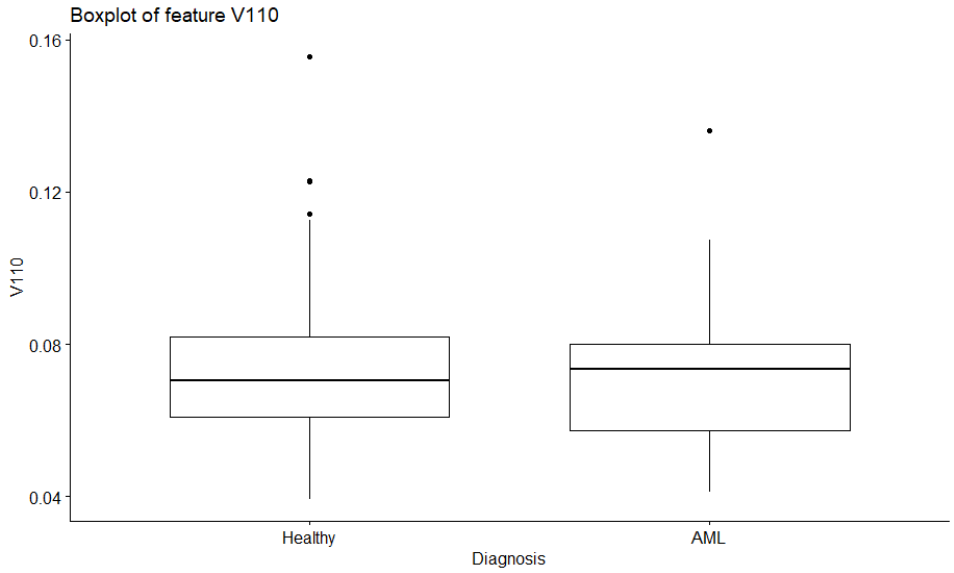

Figure 7: Boxplot of feature V110

2.2 Sub-question 1.1(b) Get a holistic view on the data

To get an idea about the complexity of the data we have embedded the data using two different methods. The first method uses the first two principal components obtained from the PCA. This method should show if there are any linear relationship between the different features. The result of this can be seen in figure 8. We see that the two different groups do not show a clear separation, instead the healthy patients are clustered closely together and the AML patients are spread out more. Since there is an overlap of the two clusters it will be more difficult to classify them later on.

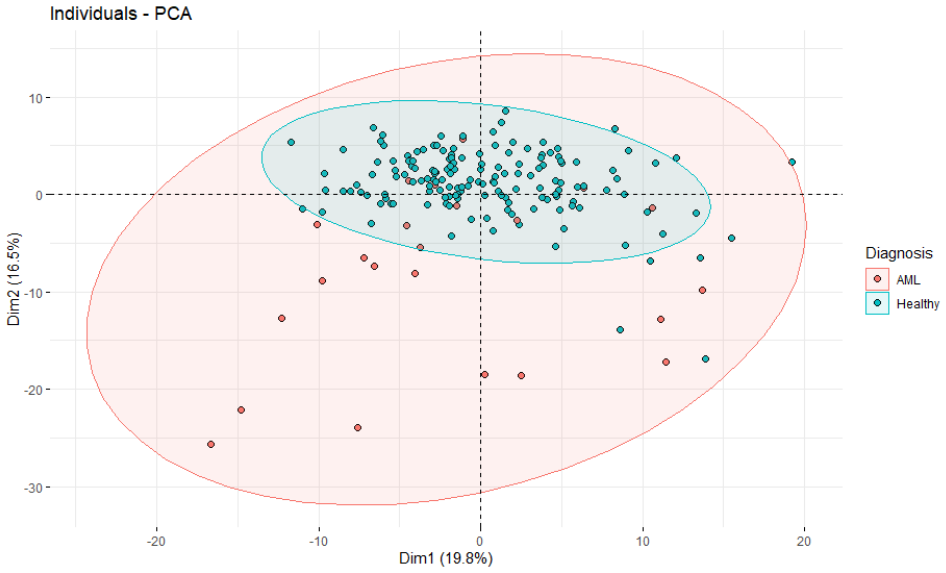

Figure 8: 2D PCA plot for the training set using the two first principal components. AML and healthy patients are shown in red and green respectively. Concentration ellipses are also drawn around the groups, which assume that the data is multivariate normal distributed

The second method uses t-SNE to embed the data. Different perplexity values between 5 and 50 were tested to see if any pattern could be observed, the clearest plots were obtained for a perplexity value of 30. Figure 9 shows a plot obtained for perplexity = 30. We observer a slight pattern where AML patients are placed on the outside of the plot and the Healthy patients near the center, however no clear seperation can be seen. This means that using the original dataset without preprocessing will likely not give good results for the classifier.

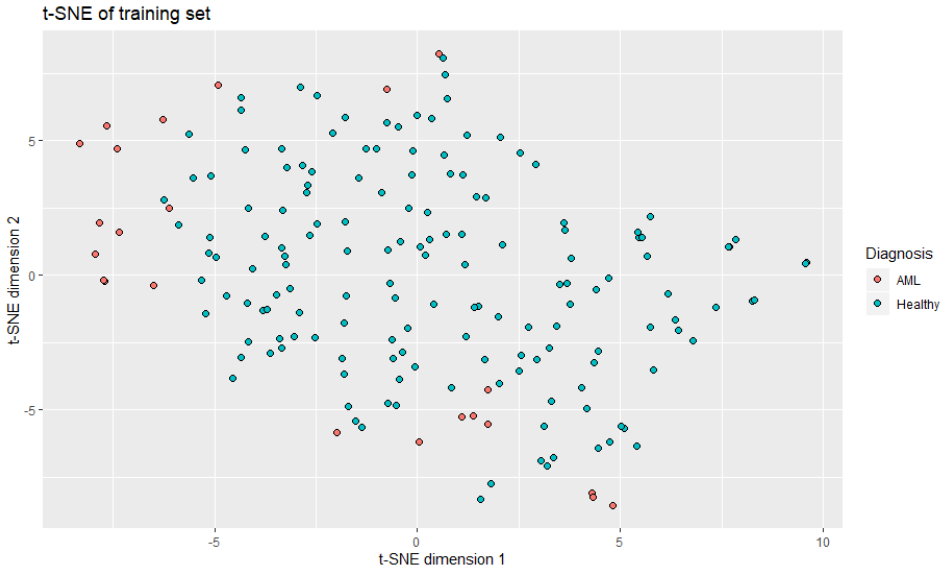

Figure 9: 2D t-SNE plot for the training set

Next we investigate the differences between the training set and test set. To do so we project the test onto the first two principal components obtained from the principal component analysis of the training set. In figure 10 the corresponding plot is shown. The test set data points are marked in blue as "Unidentified". The other data points are the labeled data points of the training set. We observe that the fast majority of the test set falls into the "healthy" cluster that we obtained from the PCA of the training set. Furthermore the distribution seems to be similar to the training set; a big central cluster with multiple, data points sporadically spread on the outsides.

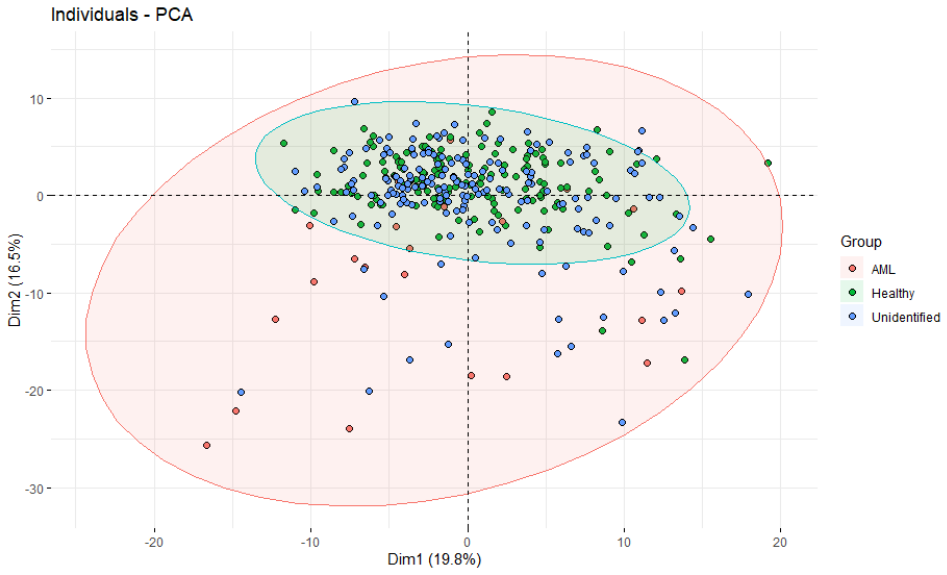

Figure 10: 2D PCA plot for the test set embedded using the principal components of the training set. The test set data points are marked in blue as "Unidentified"

2.3 Sub-question 1.1(c) General impact of preprocessing

Next we try to see the effect of different preprocessing methods on the dataset. As for the different preprocessing methods we have used feature selection, z-score transform and PCA with leading eigenvectors. The impact of preprocessing was measured with t-SNE embedding.

First we try feature selection. To find out the best number of features to select we tried feature selection with 1 to 186 features and evaluated each

based on the accuracy obtained when using kNN. The highest accuracy was achieved with 26 features (and k = 3). These 26 features are the features with the highest F values from the ANOVA analysis as the best features to use. A t-SNE plot, using a perplexity of 30, of these selected features can be seen in figure 11. Different perplexities were tested but values of around 30 seemed to have the best results. We see that a slightly better grouping has been achieved than t-SNE with the full dataset. The AML patients are clustered in the top left of the figure, and while there is still a bit of overlap between the two groups feature selection should give positive results in the classification methods.

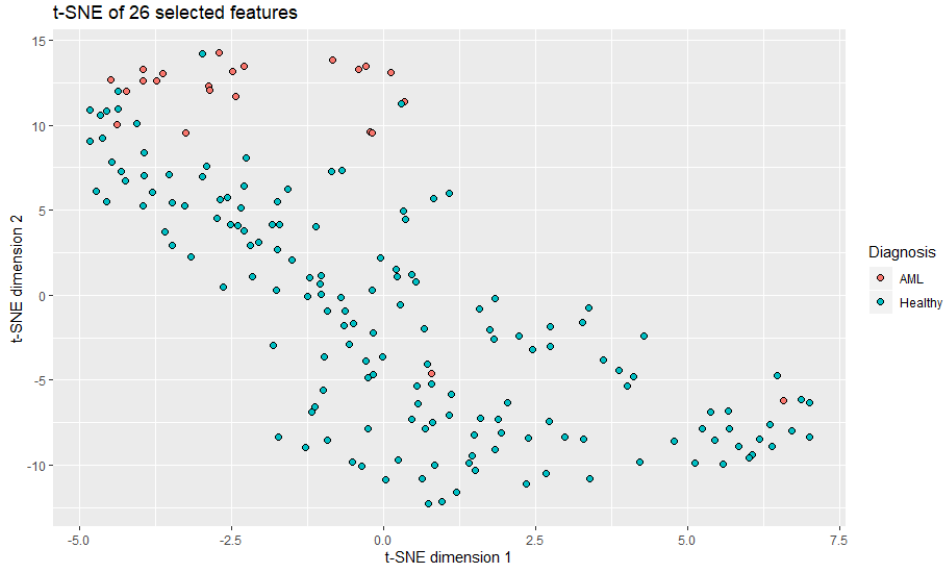

Figure 11: 2D t-SNE plot for the feature selection

Next we try z-score transform of the training set. The result of this can be seen in figure 12. Unfortunately this method of preprocessing does not show any promising results for classification since instead of forming one clear cluster, multiple small one are created. However this is not surprising since this method is very simple and does not use relations between features to try and improve grouping.

Figure 12: 2D t-SNE plot for z-score transformed training set

The final method that we have tried is PCA with leading eigenvectors. In 2.1 we determined that 25 and 38 principal components explain 90% and 95% of the variance respectively. Therefore we chose to use these numbers as our number of leading eigenvectors, along with another plot where we select 50 leading eigenvectors. Analysing the plots (shown in figure 13, 14 and 15) we observe that they all show a similar pattern: most of the AML patients are scattered around the outsides with the healthy patients in between. The patterns are quite clear and therefore we expect that leading eigenvectors preprocessing will perform well as a preprocessing method.

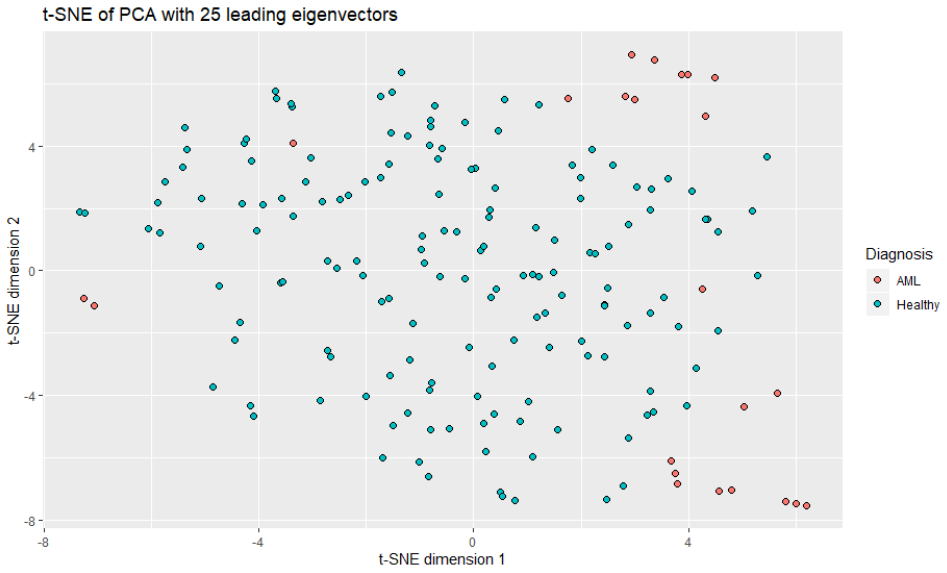

Figure 13: 2D t-SNE plot for the PCA with 25 leading eigenvectors

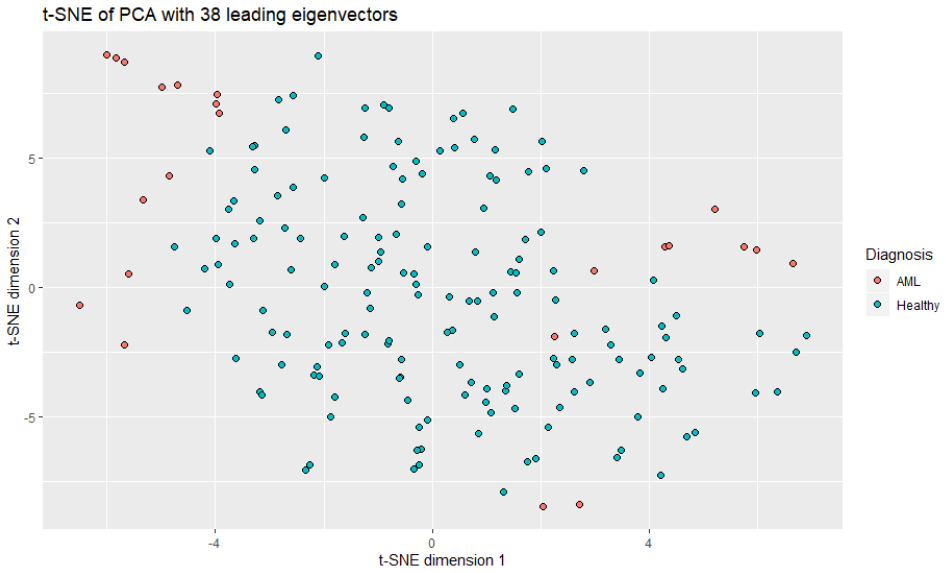

Figure 14: 2D t-SNE plot for the PCA with 38 leading eigenvectors

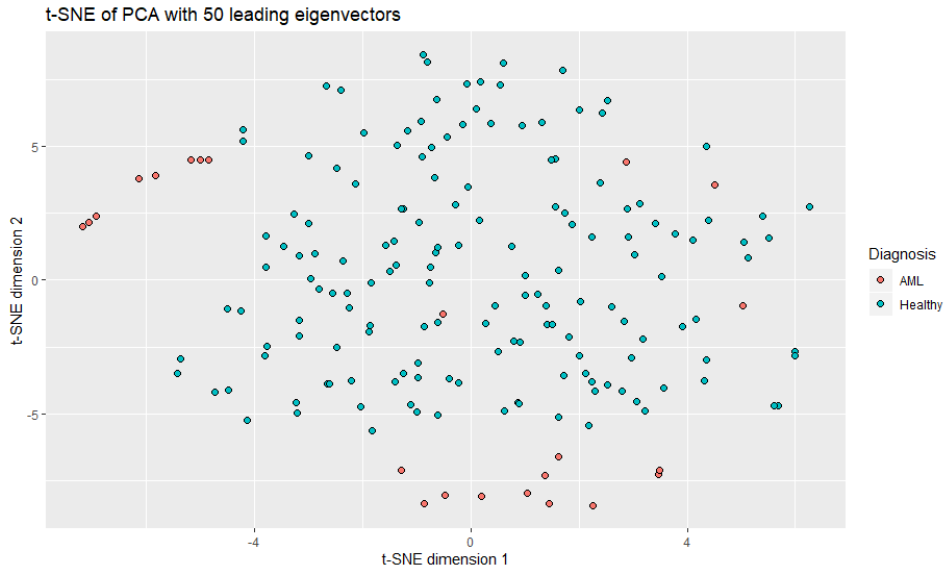

Figure 15: 2D t-SNE plot for the PCA with 50 leading eigenvectors

After analysing all the plots of preprocessed data using t-SNE embedded we noticed that one patient classified as AML was a consistent outlier, namely patient 57. In figures 16, 17, 18 and 19 patient 57 is explicitly labeled to illustrate it being a constant outlier. Because of this observation we decided to evaluate our prediction models in the next chapter on both the training set with and without patient 57.

We included t-SNE plots of the different preprocessing methods on the training set without patient 57 in appendix D. Overall these plots seem to have a minimally better separation between the two classes.

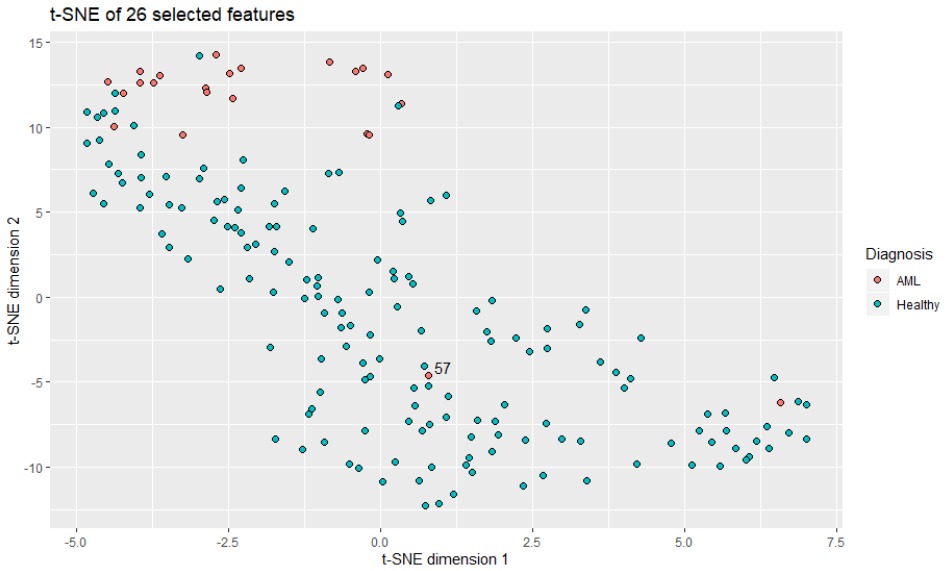

Figure 16: 2D t-SNE plot for the feature selection with patient 57 labeled

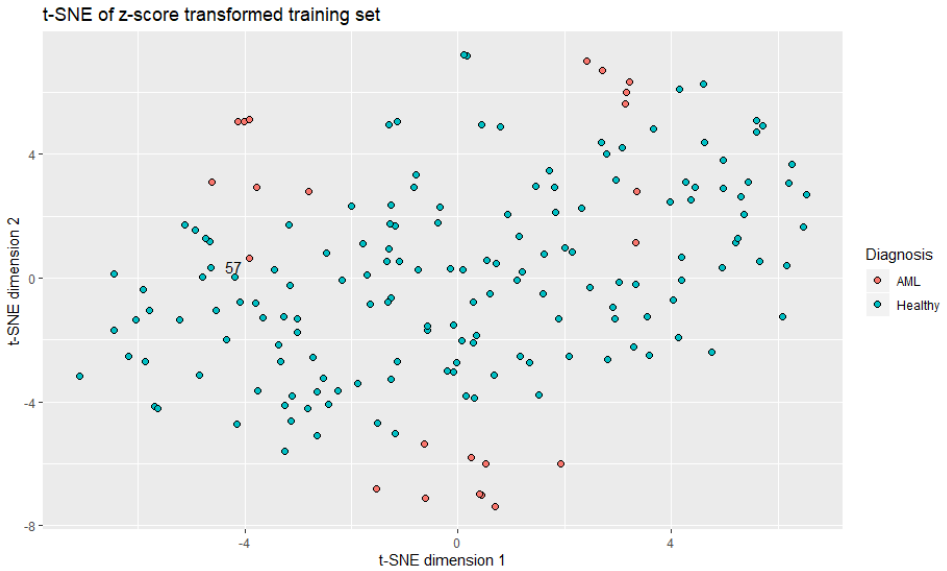

Figure 17: 2D t-SNE plot for z-score transformed training set with patient 57 labeled

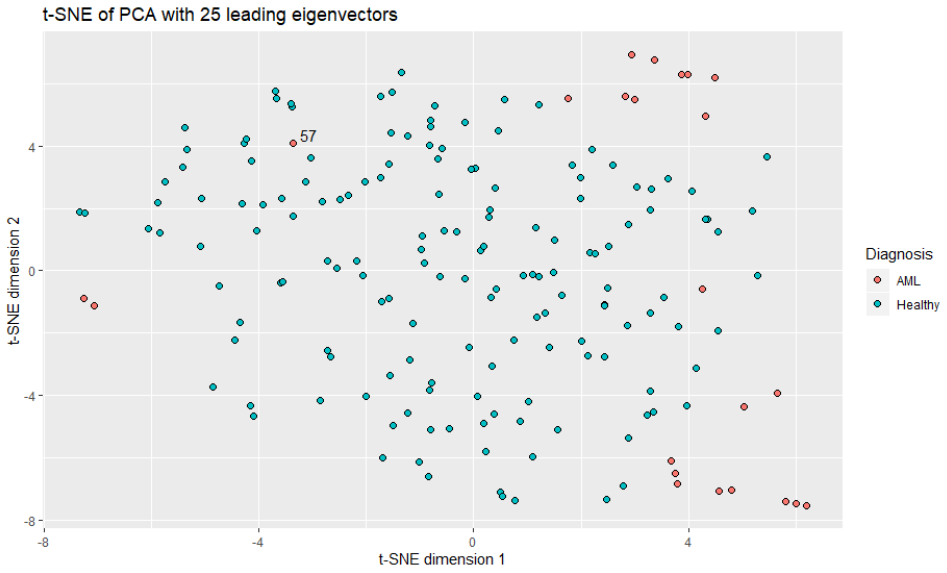

Figure 18: 2D t-SNE plot for the PCA with 25 leading eigenvectors and patient 57 labeled

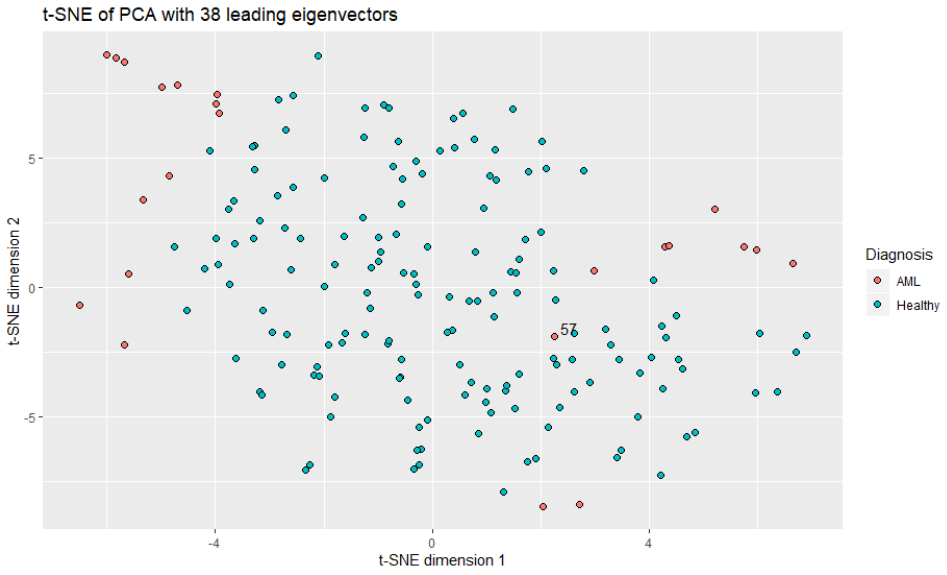

Figure 19: 2D t-SNE plot for the PCA with 38 leading eigenvectors and patient 57 labeled

3 Question 1.2 Experimentation to find the best prediction

3.1 Sub-question 1.2(a) Base-line experiments

For the base-line experiments we chose to use kNN and decision trees. In table 4 (in the appendices) the results of kNN using different preprocessing methods and different values for k are shown. These results were obtained by performing 5-fold cross validation which were repeated 10 times for better statistics. We note that the preprocessing was only based on the training set folds so that no information of the test fold leaked into the classifier. We see that the leading eigenvectors preprocessing method gives us the best accuracy and the highest kappa coefficient (accuracy corrected for the possibility of agreement occurring by chance) for k = 1. We also tried kNN on the training dataset without patient 57. Again the leading eigenvectors preprocessing method gives the best result. Excluding patient 57 resulted in higher accuracies and kappa coefficient values for both the z-score transform and leading eigenvectors methods. This further indicates that patient 57 was an outlier.

In table 6 (in the appendices), the results of decision tree-based classification using different preprocessing methods, for some values of n (number of samples required to split a tree node) are shown. Similarly to the KNN tests, an accuracy test without patient 57 was made as well. Even though the accuracy scores improved a bit, the inclusion/exclusion of this patient didn't influence the decision trees too much. What was noticeable was that the exclusion of this patient produced better results for lower values of samples per split. Since the accuracy of the Decision Tree Classifier was pretty high and immutable with different kinds of pruning, the only variable parameter was the number of samples required to split a node. Overall, since the accuracy is high and the amount of AML labels predicted is close to 20 for this classifier, overfitting doesn't seem much of an issue.

Since the t-SNE plots of leading eigenvalues and feature selection preprocessing show the clearest separation between the two classes we would expect that these methods give the best results. After performing kNN and decision tree classification it turns out that these preprocessing methods indeed perform very well. Surprisingly, while the z-score transform didn't look too promising initially, it performed really well for both kNN and decision trees. No preprocessing gives bad results because of the high dimensionality of the

data. Furthermore, as expected, low k values give better results because of the imbalanced dataset i.e. there are only 21 AML patients while there are 156 healthy patients causing a larger number of wrong classifications when $k > 1$, since then the classification is skewed towards the larger group.

Next it was checked if the models have any overfitting. We can identify overfitting by the fact the the accuracy of the training data will be much higher than that of the test data used in the cross validation. Since we used k-fold cross validation we would expect overfitting to be reduced since we test the obtained model on parts of the dataset. To further test if we suffer from overfitting we will compare the training and test error for one of the best performing methods, which is kNN with leading eigenvectors preprocessing for $k = 1$ (without patient 57). From table 4 we see that the accuracy on the test set is equal to 0.972, using the same model on the test data gives an accuracy of 1.0, since the accuracy on the training data is almost the same as the test data we can say that there is no overfitting. To further test this we have plotted a training curve, which can be seen in figure 20. We see that the training accuracy is always equal to one, which is logical since for $k = 1$ the nearest neighbour is always the point itself. However since the test accuracy is also very close to 1 for all training size we can again conclude that there is no significant overfitting.

3.2 Sub-question 1.2(b) Ensemble

In order to improve the accuracy of the prediction model without causing overfitting, an ensemble model of Decision tree classifier and KNN was created. For the ensemble, we used a sampling technique called Tomek Links, an under sampling method that looks for samples that are nearest neighbors to each other and have different labels, and removes the one with the majority label. The feature selection method was ANOVA's F values. While the accuracy of this ensemble didn't seem too exciting when compared to both KNN and DT (about 0.95492), this model was able to capture all 20 AML patients, unlike it's 2 submodels (this is a good evidence that the ensemble's classification was not affected by overfitting).
Under the exact same kind of preprocessing, Random Forest models had a slightly better accuracy, but were unable to identify 20 patients (in most cases 21 or 22 patients were identified), this is a sign of overfitting. In (appendix B) the KNN and DT ensemble is compared to the best RF model tested (n-estimators=1000):

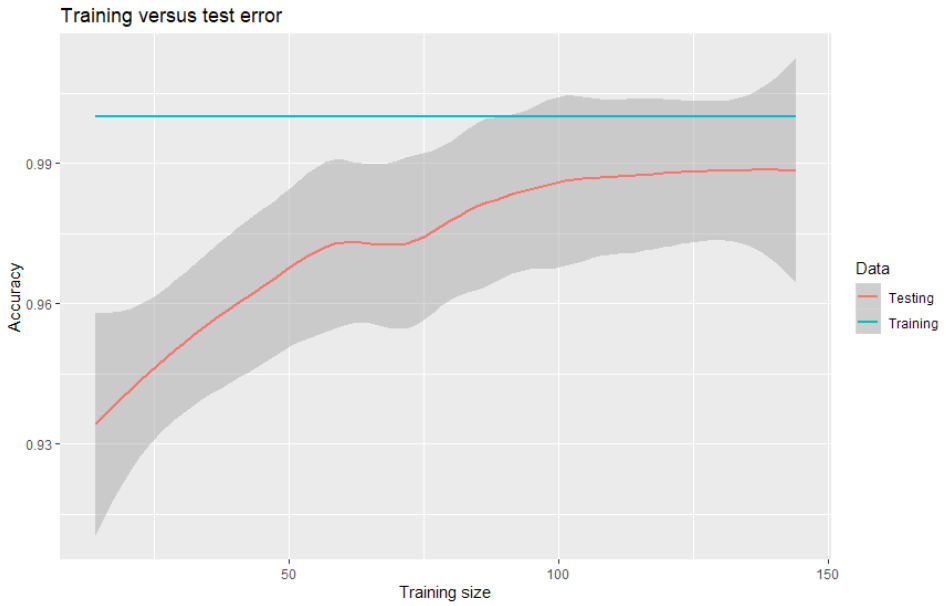

Figure 20: Graph showing the test error(orange) versus the training data(blue) for different sizes of the training set

Classifier	Accuracy	Error	Nº of identified AML patients
Random Forest	0.97206349	0.03	21
KNN and DT Ensemble	0.95492063	0.02	20

Table 2: RF vs KNN and DT Ensemble

3.3 Sub-question 1.2(c) Summary of your experiments

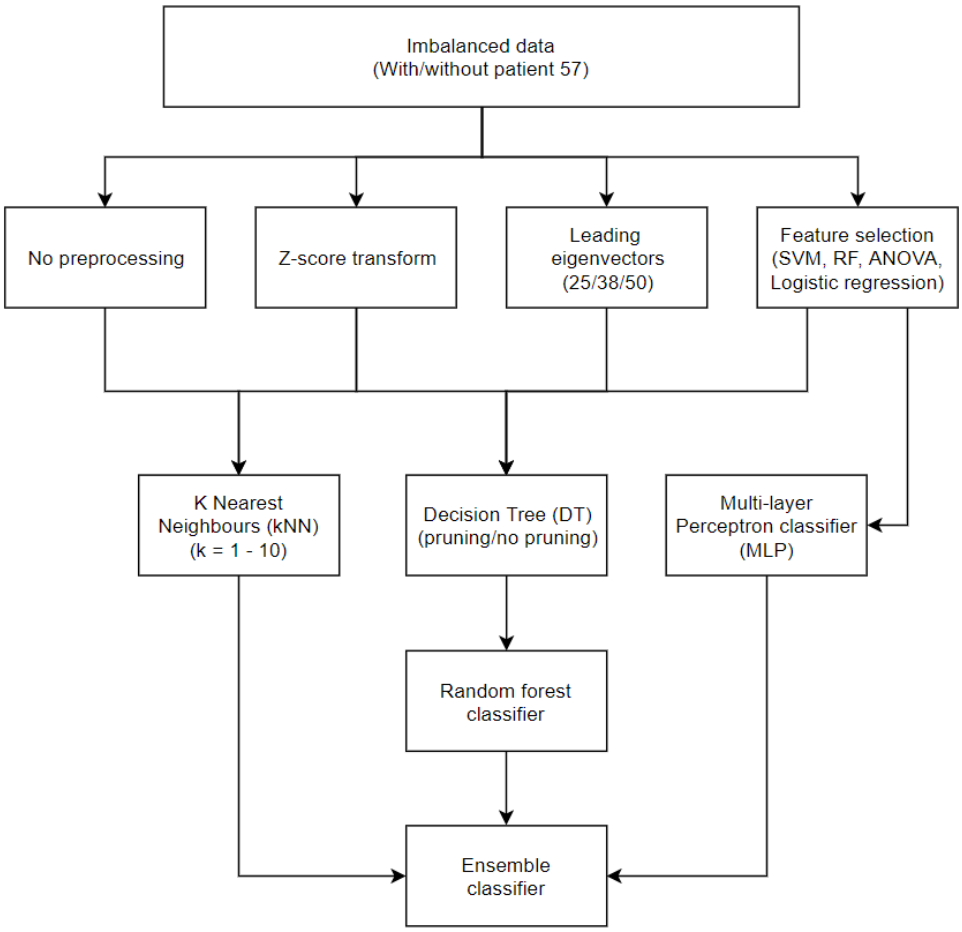

Figure 21: Summary of experiments

After the Ensemble classifier, a number of combinations of sampling methods, feature selection methods and classifiers were compiled and visualized in figure 21 above. For the sampling, 3 different methods were used: Random Over Sampling (ROS), that replicates random samples from the minority class, Random Under Sampling (RUS), that deletes random samples from the majority class and Tomek Links (TL)(explained in 1.2b). For the feature selection, we used Support vector machines (SVM), Extra tree-classifiers

(ET), Logistic Regression, ANalysis Of VAriance (ANOVA) and Random Forest (RF). Finally, we tested different classifiers (besides KNN and Decision tree that were analysed in 1.2(a)): Random Forest Classifier (RF), Support Vector Machine Classifier (SVM), a Multi-Layer Perceptron classifier (MLP) (which is a neural network that optimizes the log-loss function using LBFGS) and an ensemble of the methods mentioned above adding the KNN classifier as well.
The most promising combinations are displayed in table 7 in appendix C. One should notice that the choice of the best performers was based not only on the k-Fold Cross validation Accuracy, but on the ability to identify the 20 AML patients as well. This was a good way to check if a certain model was overfitting, since there were cases of combinations of methods that did really well in the cross validation accuracy, but failed in identifying 20 patients (e.g.: RF Classifier w/ROS and ET feature selection: 0.9935 in accuracy but identified 29 patients).

Bibliography

[1] LabTestsOnline. Flow cytometry. https://labtestsonline.org/flow-cytometry. Accessed: 2019-30-09.

[2] Kieran O'Neill, Nima Aghaeepour, Josef Špidlen, and Ryan Brinkman. Flow cytometry bioinformatics. *PLoS Comput Biol*, 9(12), 2013.

Appendix A

	No preprocessing		Z-score transform		Feature selection		Leading eigenvectors	
k	Accuracy	Kappa	Accuracy	Kappa	Accuracy	Kappa	Accuracy	Kappa
1	**0.9237563**	**0.5639986**	**0.9666229**	**0.8254220**	0.9501330	0.7710574	**0.9716396**	**0.8523376**
2	0.9203286	0.5307110	0.9570665	0.7604625	0.9388631	0.7299120	0.9627014	0.7962834
3	0.9136135	0.4384622	0.9521424	0.7302813	**0.9668082**	**0.8471294**	0.9610498	0.7852070
4	0.9096744	0.4055940	0.9392853	0.6412976	0.9501330	0.7656949	0.9465075	0.6934766
5	0.9069125	0.3804018	0.9292651	0.5597432	0.9556885	0.7859373	0.9364732	0.6202809
6	0.8968631	0.2776886	0.9214547	0.5037158	0.9448777	0.7068935	0.9253595	0.5383304
7	0.8913059	0.2284778	0.9086577	0.3845006	0.9391634	0.6664027	0.9159442	0.4583560
8	0.8874153	0.1848430	0.9025131	0.3297980	0.9445689	0.7099362	0.9109108	0.4135855
9	0.8845899	0.1511731	0.8997194	0.3075418	0.9502831	0.7504270	0.9058331	0.3717762
10	0.8845899	0.1511731	0.8997194	0.3061926	0.9499743	0.7611599	0.9024981	0.3307576

Table 3: Results of kNN **with** patient 57

	No preprocessing		Z-score transform		Feature selection		Leading eigenvectors	
k	Accuracy	Kappa	Accuracy	Kappa	Accuracy	Kappa	Accuracy	Kappa
1	**0.9294809**	**0.5576424**	**0.9736315**	**0.8609297**	**0.9376190**	**0.7420876**	**0.9747585**	**0.8678462**
2	0.9260056	0.5372606	0.9640909	0.7974713	0.9263492	0.6674282	0.9703140	0.8404777
3	0.9137314	0.4018915	0.9601862	0.7725419	0.9319048	0.7109435	0.9725212	0.8503674
4	0.9114607	0.3741432	0.9462145	0.6736405	0.9374603	0.6985829	0.9579172	0.7511152
5	0.9046830	0.3083090	0.9299103	0.5473989	0.9206349	0.6311744	0.9421845	0.6465978
6	0.8985084	0.2498390	0.9187658	0.4578383	0.9206349	0.6311744	0.9304985	0.5579985
7	0.8962544	0.2313561	0.9120347	0.3840047	0.9150794	0.6007126	0.9198619	0.4673756
8	0.8912218	0.1735488	0.9047632	0.3130820	0.9150794	0.6007126	0.9109078	0.3758722
9	0.8878250	0.1359430	0.9030180	0.2968739	0.9150794	0.5938849	0.9080815	0.3486472
10	0.8883805	0.1419624	0.9030648	0.2975420	0.9322222	0.6752821	0.9075418	0.3432723

Table 4: Results of kNN **without** patient 57

Appendix B

	No Preprocessing	Z Score	Feature Selection	Leading Eigenvectors
n	Accuracy	Accuracy	Accuracy	Accuracy
2	0.933333333	0.968225351	0.973109356	0.972521587
3	**0.965740741**	0.975726715	0.973284712	0.968553333
4	0.935648148	0.976489841	0.971758461	0.962471058
5	0.928240741	0.959061956	0.984302341	0.94792032
6	0.908333333	**0.990384615**	0.98218832	0.960792181
7	0.948148148	0.974427169	0.980837425	**0.977892084**
8	0.934721091	0.966422466	**0.986868351**	0.958174822

Table 5: Results of DT **with** patient 57 (n is the number of samples required to split a tree node)

	No Preprocessing	Z Score	Feature Selection	Leading Eigenvectors
n	Accuracy	Accuracy	Accuracy	Accuracy
2	**0.971296296**	**0.988982372**	0.994522319	**0.979694969**
3	0.943981481	0.975087597	0.993589744	0.973664043
4	0.953240741	0.987507469	0.984302341	0.972521587
5	0.944907407	0.976314484	0.996031746	0.954833915
6	0.937037037	0.978344074	**0.996794872**	0.965472212
7	0.940277778	0.982572115	0.981860338	0.953307663
8	0.95462963	0.973024925	0.977632297	0.977201618

Table 6: Results of DT **without** patient 57 (n is the number of samples required to split a tree node)

Appendix C

Sampling Method	Feature Selection	Classifier	Accuracy	Error	Nº of AML patients identified
TL	Regression	Ensemble	0.949563	0.03	20
TL	ANOVA	Ensemble	0.948139	0.03	20
TL	Regression	SVM	0.949365	0.03	20
ROS	SVM	RF	0.987097	0.01	20
TL	ANOVA	SVM	0.949365	0.03	20

Table 7: Best performers

Appendix D

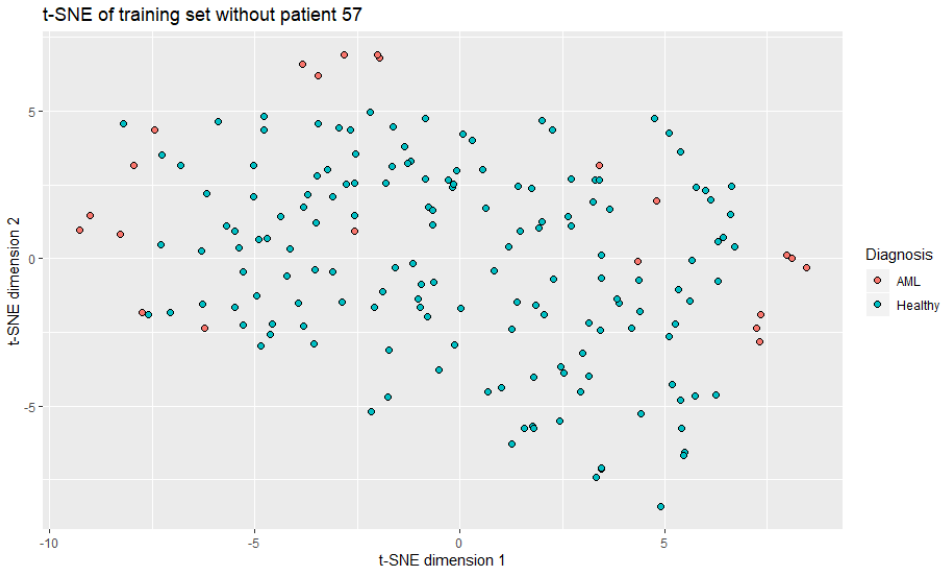

Figure 22: 2D t-SNE plot for the training set with patient 57 excluded

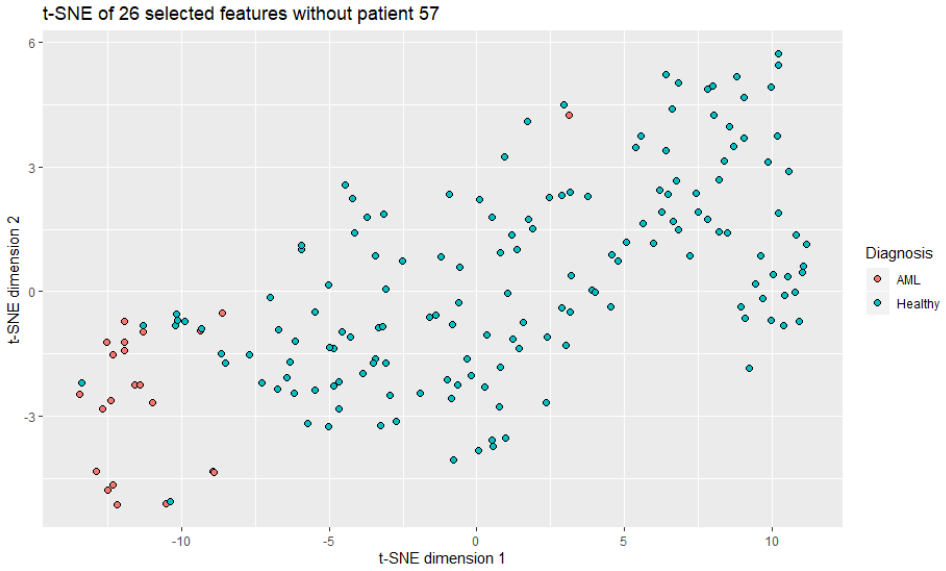

Figure 23: 2D t-SNE plot for the feature selection with patient 57 excluded

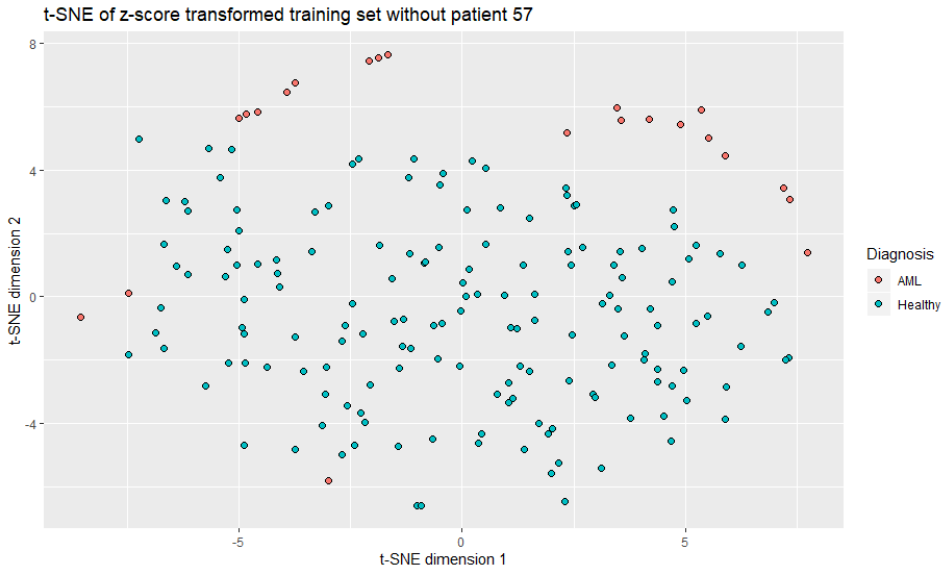

Figure 24: 2D t-SNE plot for z-score transformed training set with patient 57 excluded

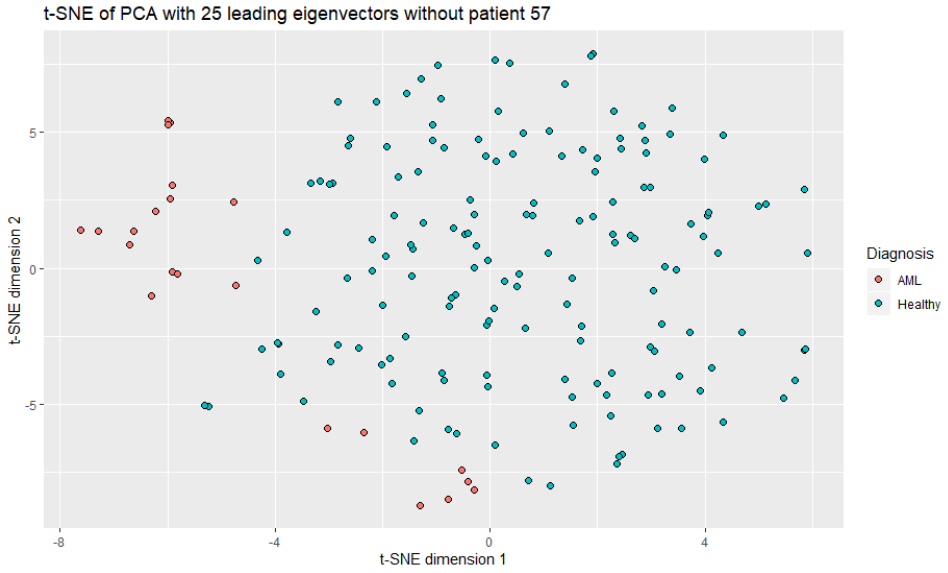

Figure 25: 2D t-SNE plot for the PCA with 25 leading eigenvectors and patient 57 excluded