Identification of Glioma Grade Using Machine Learning

Fabio Blom (4556089), Sterre de Jonge (4477464), Esther van Marrewijk (4531450), Janno Schouten (4476565)

April 11th 2020

Link GitHub: `https://github.com/JSSchouten/TM10007_Group_10.git`

1 Introduction

Gliomas are the most frequent type of intracranial tumor, comprising 81 % of the diagnosed brain malignancies. [1] The location of gliomas varies and the malignant tissue differs in cell type and grade. The Glioma Grade is considered to be an important prognostic factor and ranges from I to IV. Grades I and II are considered low grade glioma (LGG) and have a relatively high survival rate (approximately 47 % over 10 years). A glioma with a III-IV Grade is defined as high grade glioma, or glioblastoma multiforme (GBM) and is much more aggressive, which results in poorer prognosis (median survival of 15 months). Furthermore, Glioma Grade is also an important factor for decision to proceed with surgery. Surgical removal of the glioma is commonly used as treatment, but the surgical procedure depends, among other things, on the Glioma Grade.[2]

The grade may be determined with Magnetic Resonance Imaging (MRI). Varying results have been reported concerning diagnosis of gliomas. I.e. a high classification has been found with MRI for differentiating GBM from LGG, resulting in a sensitivity of 0.81 and 0.87. [2] However, MRI is still not considered as the optimal 'method' for diagnoses of gliomas. Research has shown that biological specificity of the MRI signal is poor and even expert radiologists have difficulty recognizing the tumor tissue. [3] In this case biopsy of the Glioma tissue is needed for histologic examination. It would however be preferable that biopsy is avoided since this procedure is invasive for the patient. For this reason, a Machine Learning Glioma classifier is developed.

The aim of this research is to differentiate between the tumor grade types LGG and GMB based on features extracted from multiple MRI scans (T2-weighted, T2-weighted FLAIR and T1-weighted before and after administration of a contrast agent) by means of a machine learning classifier.

2 Methods

The data was collected in a multi-centre trial from 167 patients. The data obtained from the MRI scans contained two data sets labeled respectively as LGG (Low Grade Glioma) and GBM (GlioBlastoma Multiforme). The GBM dataset contained 102 samples and 725 features and the LGG dataset 65 samples and 725 features. These data sets were combined. The features included information about intensity, volumetric, morphological, histogram-based, and textural parameters obtained from MRI-scans, as well as spatial information and parameters extracted from glioma growth models. [4] The relatively high amount of features compared to the samples caused a dimensionality problem.

2.1 Experimental and evaluation set-up

In the experimental set-up the data was firstly split in a training and test set with a distribution of 80 % and 20 % respectively. Consequently, all preprocessing steps and model optimization is trained on the training set and thereafter applied on the test set for model evaluation.

A grid search was performed for model optimization. Two feature selection methods, combined with three classifiers, were evaluated using five-fold cross validation. Performance was assessed using the mean Area Under the Curve (AUC) of the Receiver Operating Characteristics (ROC) of the five fold cross validation. The best performing combination of the feature selection method and classifier within the grid search results in one best performing method which is evaluated in model evaluation.

Evaluation was performed by fitting the selected classifier on the training set, after feature selection using the selected method. Performance was expressed using the AUC, as well as the sensitivity and specificity of the classifier. To adjust for random errors caused by different data splits, this whole process of splitting the data, fitting the selected classifier and evaluating its performance, was iterated 100 times. Final primary outcome was the mean AUC, sensitivity and specificity of these iterations. When applicable, depending on the selected feature selection method, secondary outcome would be the prevalence of the selected features in all iterations. A pair plot was made to show the distribution of the data with five features or components that had the highest prevalence.

2.2 Preprocessing

2.2.1 Data Adjustments

Firstly, the data was divided into a training and test set with a distribution of 80 % and 20 %, respectively. The data contained non-numeric values, while numeric values are required for classifiers to be operable. Possible non-numeric values that cause errors are NaNs (not a number), strings (words), and infinite numbers. To prevent errors, these values needed to be replaced. Firstly, strings and infinite numbers were replaced by NaN. Then, features, for which over 50 % of the samples contained NaN, were excluded from further classification. In both the training and test set, remaining NaNs were replaced with the median value of that feature in the training set.

2.2.2 Scaling

Thereafter, standard scaling was applied to all features. In this method, all feature values are scaled to normalized values; values ranging from zero to one. A disadvantage of this method is that outliers will dominate the minimum or maximum end of the range and thus will influence the distribution of the rest of the data. Outliers will therefore probably cause the feature to not be selected in the feature selection. With another scaling method, for instance with robust scaling, this will not occur. However, it depends on the data distribution within features which scaling method is most optimal. Since our dataset consists of 725 features we decided that it does not matter to make a decision based on data distribution.

2.2.3 Model optimization

The model was optimized using a grid search method. For each feature selection method and classifier, different values for the most important hyperparameter(s) was/were chosen and set. These values are summarized in Appendix A.1. After that, for all combinations of feature selection methods and classifiers with all values for the subsequent hyperparameters, models were trained on the training datasets of the cross validation. Using the validation datasets, the performances of the models were analyzed, with the AUC as a measure for performance. After cross-validation, the mean AUC performance over all validation datasets was used for further analysis. The best feature selection method was chosen based on the mean performance over all classifiers. The best classifier was the classifier that would be trained for the model evaluation with the test dataset and was the one with the highest performance when using the chosen feature selection method. This model optimization was performed once, based on one split in the data. In the model evaluation, the model with the best combination of feature selection and classifier with given hyperparameters was trained and tested multiple times with multiple data splits.

2.2.4 Feature selection

As mentioned before, the data consists of many more features than samples. Too many features is unwanted because that will lead to overfitting. A feature selection method has to be applied in order to derive a small amount of features that can differentiate between the classes the best. Two possible feature selection methods were evaluated: principal component analysis (PCA) and univariate statistical testing.

PCA was applied to extract a certain amount of components that contain the most variance in the data. This hyperparameter, the accumulated amount of variance that is covered with the chosen components, was evaluated using 0.5, 0.75, 0.9 and 0.95 as thresholds in the grid search. On the one hand, much variance is required because the dataset may best be classified with the components that contain most variance. On the other hand, it is unwanted to select too many components. Therefore, it was decided to use this range.

Univariate statistical testing was applied to extract a certain amount of features that statistically obtain the highest score for predicting a class, in this case the tumor grade. The ANOVA F-value was used for this method, since this test compares numeric data for multiple samples. In this test, the variability between group means is compared to the variability within groups to determine if these means are statistically different. The hyperparameter, the amount of features to select with the best ANOVA F-value, was set to 5, 10, 25, 50 and 100 in the grid search. Selecting more than 100 features was considered not to be optimal, since then the amount of features would be greater than the amount of samples in the training dataset.

2.2.5 Classifier

Finally, for the classification of the data into either LGG or GBM, different algorithms were used for model optimization. The hyperparameters of these algorithms were optimized in a grid search with the validation dataset, as described above. The classifiers Support Vector Machine (SVM), k-Nearest Neighbors (kNN) and Random Forest (RF) were analyzed. It was chosen to evaluate one simple classifier and two more complex classifiers to assess which classifier best suits the complexity of the data.

In the RF classifier only the number of trees was varied, since this is considered the only reliable hyperparameter for tuning. Hereby the applied numbers were 10, 30, 100 and 200, since these are commonly used values [5]. For hyperparameter tuning of the SVM classifier there was varied in a Linear, Polynomial and radial basis function Kernel. Furthermore, different amounts of slack were tested, varying between 0.05, 0.1 and 0.3. Finally, for the kNN classifier the number of Neighbours was altered between 3, 7, 11 and 15 neighbours because when dealing with classification between two classes, often odd numbers are used in the range 3 till 15 [6]. For all hyperparameters of the classifiers, except the kernel in SVM, hold that different values were chosen to reach different levels of regularization. For RF and kNN more regularization is reached for higher numbers of trees and neighbours and for SVM more regularization is reached for smaller values of slack.

3 Results

3.1 Model Optimization

In this section the results will be discussed for one particular run of the model that results in one best performing method. It should be noted that different results are obtained for a different run of the model, the method described here is therefore not the "only" best performing method. In Appendix A.2 the results are shown for one run. Based on these results the model is optimized. In Table 1 the results are summarized and the mean performance over all the classifiers is shown for each different feature selection method. There is not much difference visible between each feature selection method, the only notable difference is visible between PCA and univariate feature selection in which the mean performance for univariate feature selection is higher. Across univariate feature selection little difference is seen. With a selection of five features, an AUC of 94 % is achieved. The feature selection method that achieves the highest mean performance is univariate feature selection with 25 features, equal to 95 %. This results in the first hyperparameter optimization. Within this feature selection method the maximum AUC performance score of 98 % is reached with RF with 100 trees (Appendix A.2).

The result after applying the grid search therefore leads to the following best model: feature selection with univariate (25 features) in combination with RF as classifier (100 trees).

Table 1: Mean performance over classifiers for different feature selection methods

Feature Selection method	Mean performance (AUC) [%]
PCA 0.50	0.839
PCA 0.75	0.875
PCA 0.90	0.877
PCA 0.95	0.876
Univariate 5	0.938
Univariate 10	0.942
Univariate 25	0.950
Univariate 50	0.941
Univariate 100	0.939

3.2 Model evaluation

The best performing model is consequently evaluated. In Appendix A.3 the results are shown for this evaluation. In table 2 these results are summarized. The mean AUC that is reached with this model over 100 iterations is 96 % and this AUC is reached with a mean sensitivity and specificity of 93 % and 84 %, respectively.

Table 2: Results over all iterations (n=100)

Outcome measure	mean value (min, max) [%]
AUC	96 (88, 100)
Sensitivity	93 (79, 100)
Specificity	84 (64, 100)

Within each iteration, there are twenty five features that are selected. In Appendix A.4 the prevalence of all the selected features are depicted, in total 81 features are selected of which eleven features are picked during every iteration. Nine out of these eleven features are volumetric parameters and are therefore volumes that describe different aspects of the tumour. Other type of features that are often chosen are textural parameters. To visualize the data distribution of a few selected features, a pair plot is made that shows the distribution of data for five features, shown in A.5. The features that are used for this pair plot, are the following: Volume ET (enhancing part) over TC (tumour core), Volume NET (non-enhancing part) over WT (whole tumour), Volume ET over WT, Volume NET over Brain, and Volume NET over TC. These features are highly discriminating because the mean AUC between five and twenty five features does not differ very much.

4 Conclusion

It can be concluded that the best performing model after model optimization obtains good mean AUC, mean sensitivity and mean specificity after evaluation. The results furthermore show that the choice in feature selection method is more relevant than the choice in classifier. Within model optimization similar AUC values are reached for different classifiers. A different run will therefore lead to a different distribution over the classifiers and a different choice in classifier.

Secondly, it can be concluded that a high performance is obtained for five features and that twenty features more only lead to an increase in performance of 1,2 %. There are therefore few features that are highly predictive.

5 Discussion

5.1 Model Interpretation and Improvements

5.1.1 Scaling

The decision was made to use standard scaling, since this scaling method is the standard in general conditions. The standard scaling did seem applicable for this data distribution, since high scores were found in the results.

5.1.2 Model optimization

The choice of feature selection and classification with the accessory hyperparameters were based on performance (mean AUC) in the validation set. Results showed a very small difference in performance, therefore it could not be concluded whether the chosen combination actually performed significantly better than others. For example, the AUC did increase in feature selection methods when more features were included. However, this difference in score was not outstanding. Furthermore, the results varied very little for different classifiers, this suggests that classifiers do not have much influence on the differentiation of this dataset. The amount of features is more valuable for further research, since we think that most improvements could be made here. In addition, only one run was performed to find the ideal model combination. Consequently, due to the fact that the differences in scores between model combinations are small and it is unsure whether they are significant, it might well be that a new run results in a different feature selection and classification method. Finally, for every feature selection and classification method only specific hyperparameters were taken in the grid search for tuning. For example, it is generally advised that for RF only the number of trees is modified. Other hyperparameters may cause overfitting. However, inclusion of more hyperparameters for other classifiers may give different outcomes.

5.1.3 Model evaluation

The fact that many model combinations gave a high AUC score in the validation set, suggests that the training data is relatively easily separable, regardless of feature selection and classifier. In line with this, the applied model method also resulted in high AUC, sensitivity and specificity scores in the test set. However, the mean values showed a large range of minimum and maximum scores. This could imply that the model is not very robust and sometimes does not perform well on the dataset. On the other hand, the minimum values could merely be outliers. To verify, statistical testing should be performed.

5.2 Clinical Relevance

The mean AUC of 96 %, mean sensitivity of 93 % and mean specificity of 84 %, imply that machine learning is a promising possibility for distinguishing GBM from LGG. The current practice sensitivity and specificity of glioma grade prediction are 81 % and 87 %. [2] Even though the model sensitivity and specificity has some variation (respectively 79-100 % and 64-100 %), the same holds for performance in radiologists, with for example the sensitivity for glioma grading ranging from 55.1 % to 83.3 % [7].

Furthermore, there were several features with a prevalence higher than 90 %. Features that are predictive are features that describe either the enhancing part of the tumor (ET) or the non-enhancing part of the tumor (NET). It is hypothesized that ET represents regions where there is leakage of contrast through a disrupted blood-brain barrier, which is commonly seen in high grade gliomas. [4] Features that describe this volume may therefore discriminate the data well. Further research needs to be done in order to research this hypothesis and to gain more insight in why some features are predictive. This insight may be beneficial for radiologists since it may eventually lead in a reduction of their workload.

5.3 Recommendations

This machine learning algorithm is not yet suited for clinical implementation. To achieve this, multiple steps need to be taken first. The RF classifier in combination with univariate feature selection performed best, but more data and statistical analysis is necessary to confirm this. For further research we suggest to focus on the amount of features, apply statistical testing to the results and evaluate the model to determine the optimal amount of features to differentiate the data that reaches the highest AUC. Additional research can be done on the features itself, to evaluate why these features are predictive.

References

[1] Ostrom QT, Bauchet L, Davis FG, Deltour I, Fisher JL, Langer CE, et al. The epidemiology of glioma in adults: a "state of the science" review. Neuro-oncology. 2014 Jul;16(7):896–913.

[2] Wang QP, Lei DQ, Yuan Y, Xiong NX. Accuracy of ADC derived from DWI for differentiating high-grade from low-grade gliomas: Systematic review and meta-analysis. Medicine (Baltimore). 2020 Feb;99(8):e19254.

[3] Upadhyay N, Waldman AD. Conventional MRI evaluation of gliomas. Br J Radiol. 2011 Dec;84 Spec No 2:S107–111.

[4] Bakas S, Akbari H, Sotiras A, Bilello M, Rozycki M, Kirby JS, et al. Advancing The Cancer Genome Atlas glioma MRI collections with expert segmentation labels and radiomic features. Sci Data. 2017 09;4:170117.

[5] Ramon J. Question Forum, answer from Ramon J from KU Leuven [Internet]. ResearchGate; 2013. [cited 01-05-2020]. Available from: https://www.researchgate.net/post/How_to_determine_the_number_of_trees_to_be_generated_in_Random_Forest_algorithm.

[6] Pavin H, Alizadeh H, Minati B. A Modification on K-Nearest Neighbor Classifier. Global Journal of Computer Science and Technology. 2010;10(14):37–41.

[7] Law M, Yang S, Wang H, Babb JS, Johnson G, Cha S, et al. Glioma Grading: Sensitivity, Specificity, and Predictive Values of Perfusion MR Imaging and Proton MR Spectroscopic Imaging Compared with Conventional MR Imaging. American Journal of Neuroradiology. 2003;24(10):1989–1998.

A Appendix

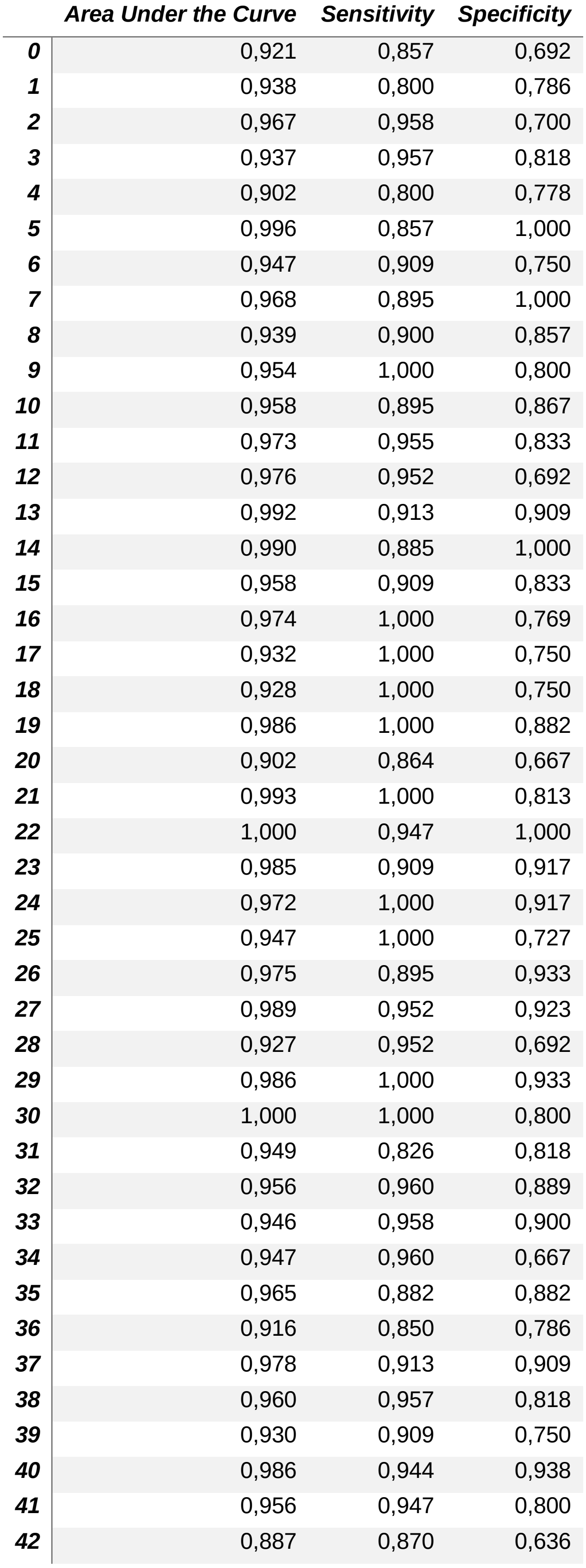
A.1 Hyperparameters

	Method	**Hyperparameter [values used in grid search]**	**Value in optimized model**
Pre-processing	Replace infinites	Replace by	NaN
	Replace strings	Replace by	NaN
	Drop NaN values	Axis to delete	Columns (feature)
	Drop NaN values	Threshold	0.5 * number of samples
	Impute NaN values	Impute with	Median
Scaling	Standard scaler	-	-
Feature selection	Principal Component Analysis	Amount of variance [0.5, 0.75, 0.9, 0.95]	-
	Univariate statistical testing	Statistical test	F-ANOVA
	Univariate statistical testing	Number of features [5, 10, 25, 50, 100]	25
Cross validation	Cross validation	Number of cross validations	5
Classifier	Support Vector Machine	Kernel [‘linear’, ‘rbf’, ‘poly’]	-
	Support Vector Machine	Slack parameter [0.05, 0.1, 0.3]	-
	K Nearest Neighbors	Number of neighbors [3, 7, 11, 15]	-
	Random Forest	Number of trees [10, 30, 100, 200]	100
Evaluation	Split data with random selection	Test size	20%
	With test set	Number of iterations	100
	Performance	Score	Area Under the Curve

A.2 Grid search results

Classifier	***PCA 0,5***	***PCA 0,75***	***PCA 0,9***	***PCA 0,95***	***Univariate 5***	***Univariate 10***	***Univariate 25***	***Univariate 50***	***Univariate 100***
RF 10	0,85	0,81	0,77	0,79	0,93	0,94	0,96	0,95	0,96
RF 30	0,86	0,89	0,84	0,86	0,93	0,96	0,97	0,96	0,97
RF 100	0,87	0,90	0,88	0,91	0,96	0,96	0,98	0,97	0,96
RF 200	0,87	0,91	0,89	0,90	0,95	0,97	0,97	0,97	0,97
SVM linear 0,3	0,85	0,90	0,94	0,92	0,95	0,95	0,96	0,93	0,92
SVM linear 0,1	0,85	0,92	0,94	0,92	0,95	0,95	0,96	0,94	0,93
SVM linear 0,05	0,84	0,91	0,95	0,92	0,94	0,95	0,96	0,95	0,94
SVM rbf 0,3	0,84	0,88	0,90	0,90	0,94	0,94	0,94	0,94	0,95
SVM rbf 0,1	0,85	0,89	0,90	0,90	0,95	0,94	0,94	0,94	0,95
SVM rbf 0,05	0,85	0,89	0,90	0,90	0,95	0,94	0,94	0,94	0,95
SVM poly 0,3	0,82	0,83	0,82	0,82	0,94	0,94	0,93	0,93	0,93
SVM poly 0,1	0,83	0,83	0,84	0,83	0,94	0,92	0,94	0,93	0,93
SVM poly 0,05	0,84	0,83	0,84	0,83	0,94	0,92	0,94	0,93	0,93
KNN 3	0,77	0,85	0,87	0,86	0,89	0,89	0,92	0,94	0,91
KNN 7	0,80	0,88	0,89	0,89	0,91	0,94	0,95	0,94	0,94
KNN 11	0,85	0,89	0,87	0,87	0,94	0,95	0,95	0,94	0,93
KNN 15	0,85	0,87	0,87	0,89	0,94	0,95	0,96	0,93	0,93

A.3 All Results



	Area Under the Curve	*Sensitivity*	*Specificity*
0	0,921	0,857	0,692
1	0,938	0,800	0,786
2	0,967	0,958	0,700
3	0,937	0,957	0,818
4	0,902	0,800	0,778
5	0,996	0,857	1,000
6	0,947	0,909	0,750
7	0,968	0,895	1,000
8	0,939	0,900	0,857
9	0,954	1,000	0,800
10	0,958	0,895	0,867
11	0,973	0,955	0,833
12	0,976	0,952	0,692
13	0,992	0,913	0,909
14	0,990	0,885	1,000
15	0,958	0,909	0,833
16	0,974	1,000	0,769
17	0,932	1,000	0,750
18	0,928	1,000	0,750
19	0,986	1,000	0,882
20	0,902	0,864	0,667
21	0,993	1,000	0,813
22	1,000	0,947	1,000
23	0,985	0,909	0,917
24	0,972	1,000	0,917
25	0,947	1,000	0,727
26	0,975	0,895	0,933
27	0,989	0,952	0,923
28	0,927	0,952	0,692
29	0,986	1,000	0,933
30	1,000	1,000	0,800
31	0,949	0,826	0,818
32	0,956	0,960	0,889
33	0,946	0,958	0,900
34	0,947	0,960	0,667
35	0,965	0,882	0,882
36	0,916	0,850	0,786
37	0,978	0,913	0,909
38	0,960	0,957	0,818
39	0,930	0,909	0,750
40	0,986	0,944	0,938
41	0,956	0,947	0,800
42	0,887	0,870	0,636

43	1,000	1,000	0,769
44	0,962	0,864	0,833
45	0,927	0,880	0,778
46	0,982	0,947	0,867
47	0,982	0,952	1,000
48	0,937	1,000	0,800
49	0,973	0,800	0,889
50	0,980	0,870	0,909
51	0,968	0,870	0,909
52	1,000	1,000	1,000
53	0,911	0,913	0,636
54	0,972	0,783	0,909
55	0,882	0,833	0,813
56	0,968	0,950	0,929
57	0,972	1,000	0,813
58	0,934	0,905	0,769
59	0,951	0,864	0,917
60	0,962	1,000	0,765
61	0,971	0,920	0,889
62	0,900	0,955	0,667
63	0,950	0,917	0,800
64	0,879	0,941	0,706
65	1,000	0,957	1,000
66	0,960	0,913	0,727
67	0,957	0,900	0,929
68	0,944	0,875	0,833
69	0,936	0,952	0,692
70	0,962	0,944	0,813
71	0,977	1,000	0,857
72	0,980	0,960	0,778
73	0,951	0,955	0,833
74	0,985	0,952	0,846
75	0,931	0,792	0,700
76	0,927	0,938	0,722
77	0,960	0,880	0,889
78	0,987	0,917	0,900
79	0,973	1,000	0,769
80	0,943	0,818	0,917
81	0,965	0,944	0,875
82	0,996	0,952	1,000
83	0,898	0,909	0,667
84	0,953	1,000	0,867
85	1,000	1,000	1,000
86	0,970	0,950	0,714

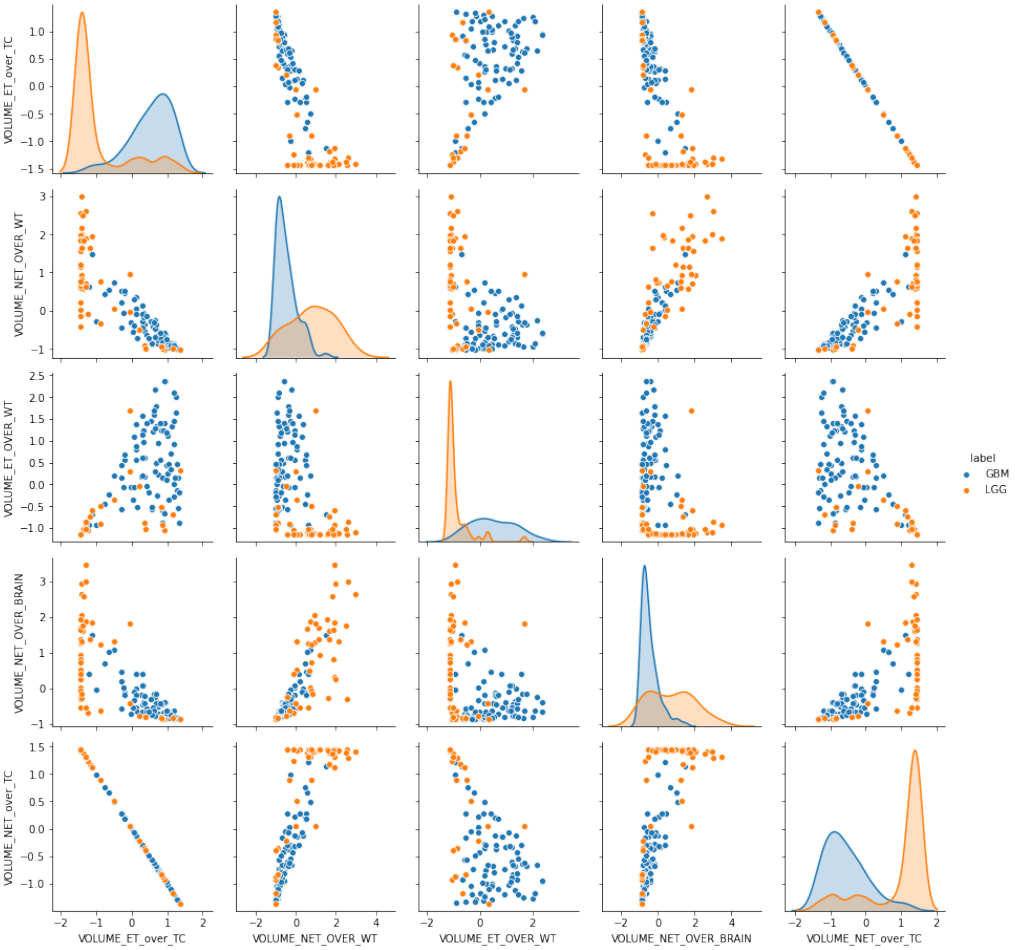
87	0,929	0,950	0,786
88	0,970	0,950	0,929
89	0,965	1,000	0,867
90	0,953	0,913	0,909
91	0,971	0,905	0,846
92	0,971	0,792	1,000
93	0,989	0,952	0,923
94	0,952	1,000	0,769
95	0,924	0,955	0,833
96	0,987	0,960	1,000
97	0,928	0,895	0,800
98	0,976	0,941	0,941
99	0,941	0,913	0,818

A.4 Selected feature prevalence

Feature	Prevalence [%]
VOLUME_NET_OVER_WT	100,00
VOLUME_NET	100,00
VOLUME_ET_over_TC	100,00
TEXTURE_GLRLM_ET_T2_LRHGE	100,00
TEXTURE_GLRLM_ET_T1Gd_GLV	100,00
VOLUME_NET_OVER_BRAIN	100,00
VOLUME_ET_OVER_WT	100,00
VOLUME_ET_OVER_BRAIN	100,00
VOLUME_ET	100,00
SOLIDITY_NET	100,00
VOLUME_NET_over_TC	100,00
TEXTURE_GLRLM_ET_FLAIR_GLV	99,00
TEXTURE_GLCM_ET_T2_AutoCorrelation	96,00
TEXTURE_GLRLM_ET_T1Gd_RLN	94,00
TEXTURE_GLRLM_ET_T1Gd_RP	94,00
TEXTURE_NGTDM_NET_T2_Busyness	89,00
TEXTURE_GLRLM_ET_T1Gd_SRE	89,00
TEXTURE_GLCM_ET_T2_SumAverage	84,00
TEXTURE_GLSZM_ET_T1Gd_ZP	80,00
TEXTURE_GLRLM_ET_FLAIR_LRHGE	77,90
TEXTURE_GLRLM_ET_T1Gd_LRE	72,00
TEXTURE_GLSZM_NET_T1Gd_SZE	55,90
TEXTURE_GLCM_ET_T2_Variance	47,00
TEXTURE_GLRLM_ET_T1_RP	25,00
TEXTURE_NGTDM_ET_T2_Busyness	24,00
TEXTURE_GLSZM_ET_T1_ZP	23,00
TEXTURE_GLSZM_NET_T1Gd_ZSN	22,00
TEXTURE_GLRLM_ED_T1Gd_LRLGE	20,00
TEXTURE_GLRLM_ET_FLAIR_RLV	19,00
TEXTURE_GLRLM_ET_T2_HGRE	19,00
TEXTURE_GLCM_ED_T1Gd_SumAverage	18,00
VOLUME_ET_OVER_ED	18,00
TEXTURE_NGTDM_NET_FLAIR_Busyness	16,00
TEXTURE_GLRLM_ET_T1_LRE	15,00
INTENSITY_STD_ET_T2	13,90
TEXTURE_GLCM_ED_T1_SumAverage	12,00
TEXTURE_NGTDM_ET_T1Gd_Busyness	11,00
TEXTURE_GLRLM_ET_T2_RLV	10,00
TEXTURE_NGTDM_ET_FLAIR_Busyness	10,00
TEXTURE_GLSZM_ET_T1_LGZE	9,00
TEXTURE_GLSZM_ET_FLAIR_LZLGE	9,00
TEXTURE_GLRLM_ET_T1_GLV	8,00
TEXTURE_GLSZM_ET_T2_GLN	6,90

TEXTURE_GLRLM_ED_T1Gd_SRLGE	6,00
TEXTURE_GLCM_ED_T1Gd_AutoCorrelation	6,00
TEXTURE_GLRLM_ET_T1_RLN	5,00
TEXTURE_GLRLM_NET_T1_LRHGE	5,00
TEXTURE_GLRLM_ET_T2_SRHGE	5,00
TEXTURE_GLCM_ED_T1Gd_Contrast	5,00
TEXTURE_GLRLM_ET_FLAIR_LRE	5,00
TEXTURE_NGTDM_ED_T1Gd_Complexity	5,00
TEXTURE_GLSZM_ET_FLAIR_LZHGE	4,00
TEXTURE_NGTDM_ED_T1Gd_Contrast	4,00
VOLUME_NET_OVER_ED	4,00
TEXTURE_NGTDM_ET_FLAIR_Strength	4,00
TEXTURE_GLCM_NET_T1Gd_AutoCorrelation	4,00
TEXTURE_GLCM_ET_FLAIR_AutoCorrelation	4,00
TEXTURE_GLCM_ET_T2_Entropy	4,00
TEXTURE_GLRLM_ET_T1_SRE	3,00
TEXTURE_GLCM_ED_T1Gd_Dissimilarity	3,00
TEXTURE_GLRLM_ET_T1_LRLGE	3,00
TEXTURE_GLSZM_NET_T1Gd_ZP	3,00
TEXTURE_GLRLM_NET_T2_LRLGE	3,00
TEXTURE_GLRLM_ET_FLAIR_LRLGE	3,00
TEXTURE_NGTDM_NET_T1_Busyness	3,00
TEXTURE_GLSZM_ET_T1Gd_SZE	2,00
TEXTURE_GLCM_ED_T1Gd_Entropy	2,00
TEXTURE_NGTDM_ET_T1_Busyness	2,00
TEXTURE_GLSZM_ED_T1Gd_ZP	2,00
TEXTURE_GLRLM_ET_T2_GLN	1,00
HISTO_NET_FLAIR_Bin8	1,00
TEXTURE_GLSZM_ET_T2_LGZE	1,00
TEXTURE_GLSZM_ED_T1Gd_SZLGE	1,00
TEXTURE_GLRLM_ET_FLAIR_SRLGE	1,00
TEXTURE_GLCM_ET_FLAIR_SumAverage	1,00
TEXTURE_GLCM_ED_T1_AutoCorrelation	1,00
HISTO_NET_T1Gd_Bin4	1,00
TEXTURE_GLCM_NET_T1Gd_SumAverage	1,00
TEXTURE_GLSZM_ET_FLAIR_LGZE	1,00
TEXTURE_GLCM_ET_FLAIR_Variance	1,00
TEXTURE_GLRLM_ET_FLAIR_RP	1,00

A.5 Pairplot



VOLUME_ET_over_TC
1.0
0.5
0.0
−0.5
−1.0
−1.5
VOLUME_NET_OVER_WT
3
2
1
0
−1
VOLUME_ET_OVER_WT
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
VOLUME_NET_OVER_BRAIN
3
2
1
0
−1
VOLUME_NET_over_TC
1.5
1.0
0.5
0.0
−0.5
−1.0
−2
−1
0
1
2
VOLUME_ET_over_TC
−2
0
2
4
VOLUME_NET_OVER_WT
−2
0
2
VOLUME_ET_OVER_WT
−2
0
2
4
VOLUME_NET_OVER_BRAIN
−2
−1
0
1
2
VOLUME_NET_over_TC
label
GBM
LGG

B Reflection

B.1 General planning, communication strategy, division of roles

From the start we frequently met up to discuss the lectures and the general assignments. Because of the current situation we did not meet up in personal, but via Teams. Communication took place mainly during Microfoft Teams sessions but also via WhatsApp. We did not start with our group assignment immediately, since we had to await certain lectures before we could effectively start. We started in the third week on the assignment. We mainly got together and worked on the code together. One person shared his/ her screen and we could all watch the changes that the person made to the code. The other team members simultaneously discussed the plan of action and the choices that had to be made. After a session together we discussed things that could be prepared for the next session, for instance parts that had to be written for the paper or some things of the code that could be investigated.

B.2 Fabio Blom

There was a very good collaboration between the group members. We often had Microsoft Teams meetings to discuss and make a plan. During these meetings we often worked on google colab together to make the python code working, by sharing screen. In this way everyone was sharing ideas about implementation of the code, while all group members were immediately informed about new adjustments. This way of working gave everyone space to give there input in the python code. However, I personally did not consider myself the best in python coding, although I do understand everything that was eventually implemented. I focused more on how several concepts had to be implemented in the general code. For example I gave my input about how the cross-validation should be elaborated and implemented in the model, while other group members, like Janno, were more focused on writing this concept in hardcore Python code. Furthermore I was involved in debating on certain important decisions in the model. In line with the elaboration on google colab, division in tasks the report of the research was well defined. Every person wrote out certain parts of the text. Thereafter, I focused on checking and rewriting the text together with Sterre and Esther. Janno and Esther made sure all the lay out of text and results were properly elaborated in LaTeX.

B.3 Sterre de Jonge

I am really positive about the teamwork within this group. I am satisfied with the result that we managed to achieve, and I enjoyed the teamwork since we worked very efficient with each other. I was a bit anxious at first that we did not start right away on the assignment, but in the end, it perfectly worked out. We mainly worked together on the code, which I found to be effective. Since we could not meet up in real-life, we communicated via Teams and WhatsApp. One team member shared his/ her computer screen and we could all watch the changes that this person made to the code. The other team members discussed the plan of action and the choices that had to be made. I am myself not the best in programming, but I am critical and precise, which I used in my contribution to the code. I spent individually quite some time on the lectures, to fully understand the principles of machine learning, this understanding I used to contribute in discussion on choices that had to be made. I enjoyed the discussions that we had, because that helped me in understanding the topics better. My contribution in this project was, furthermore, in writing on the report. I have written a part of the methods, the results and I spent time on checking and improving other peoples' work. For me this cooperation worked really well because I am very fond of effectively working together. However, a negative aspect might be that for some this cooperation does not work well because of personal preferences. For a future collaboration I think it would be better to pay more attention to this.

B.4 Esther van Marrewijk

The communication between the group members was quite good and efficient. At the beginning of the project I created a link to a Microsoft Teams meeting, so our group meetings could be held using that platform. Next, my contribution in this project consisted of creating an overview of possible ways to approach the different steps of the machine learning algorithm, based on the lectures and interpretations from the other group members. When we had this overview, we argued amongst ourselves which strategies would be most suitable for our problem. This contributed to an overall better-founded strategy and understanding of the lectures and theory. Naturally, this initial strategy had to be adjusted several times during the project, but always in good agreement from all group members and with clear reasoning. The coding itself was done during our Microsoft Teams meetings, in which one person shared his/her screen and ran the code. The other group members sought for ways to implement our strategy using available functions on the internet or from the lecture exercises. Most of the times, my contribution was in the latter, since I enjoy troubleshooting and I intuitively understand the hyperparameters of some functions after reading short explanations on the internet. Finally, the report was made in LaTeX, in which Janno and I provided the formatting of the document. Moreover, some parts of the method section and some parts of the discussion were written by me and in the end I read and improved other peoples' writings. In conclusion, a completely digital collaboration would not be my method of choice but it worked surprisingly well.

B.5 Janno Schouten

I believe that my role during this project was focused on writing the python code. This was the part of the project that I enjoy doing most, which may have led to me working on the code more than others. Writing this, I need to emphasize that I do not mean that I wrote the majority of the code by myself. Most of the code was written during video call sessions. I did spend more time than others on tidying up the code and implementing different parts of the code into functions. This was especially important for the script to be able to iterate a chosen amount of times. I spent less time in writing the report. Other group members started focusing on this part of the project earlier than I did. I did write part of the discussion together with Esther, with adjustments and additions of Fabio and Sterre. Also, I started implementing the code into LaTeX. Esther contributed with finalizing the formatting of the report in LaTeX. To finalize the report, we all read each other's writings and commented on them. In my opinion, collaboration during this project went well. Decisions were taken together, and writing of the code was performed together for the majority of the time as well. Group members which had more difficulty writing code started working on the report earlier. I enjoyed working together with this group of people and would like to do it again for another project.

B.6 Conclusion

As stated stated in all personal reflections, everyone was satisfied with the collaboration during this project. Even with the limitations caused by the corona virus, we managed to communicate well and work together as a team. We even stated that programming together might actually work better sharing screens compared to everyone working on their own screen. We did not divide different tasks with clear agreements, but this process occurred naturally during the project. In this particular project that worked well, however in future projects it might be better to divide tasks more clearly. We are all proud of the results we managed to achieve and would like to work together with this group sometime in the future.