Project Report: GBM-UPENN Data Set

Buchser Fabienne
fbuchser@student.ethz.ch

Frank Julia
jfrank@student.ethz.ch

Nagendran Varsa
vnagendran@student.ethz.ch

Sabine Schär
saschaer@student.ethz.ch

ETH Zurich, Biomedical Data Science Lab
Foundations of Data Science Project - Spring 2024

June 17, 2024

Abstract

Glioblastoma is a type of brain tumor with especially high mortality rates. The mutation status of isocitrate dehydrogenase genes (IDH1 and IDH2) is shown to correlate with the survival rates of patients and facilitates specific treatment options. Currently, the IDH mutation status can only be determined by tissue biopsy after invasive brain surgery. This project aims to train and test different machine learning models determining the IDH1 mutation status based on non-invasive clinical and radiomic features. The utilized models were Balanced Random Forest classifier, XGBoost classifier, Support Vector Machine and Logistic Regression. It has been demonstrated that mutation status can be predicted based on non-invasive features, achieving performance scores similar to those reported in existing literature. However, there is still ample room for improvement. Future studies should focus on utilizing more balanced data sets with minimal missing data.

1 Introduction

Glioblastoma is a very aggressive type of brain tumor. Of all diagnosed patients only 5% survive longer than five years. [1] Because of this high mortality rate research has been trying to utilize imaging data and machine learning to improve current diagnostic procedure. [2]

Current research has found a correlation between the mutation status of isocitrate dehydrogenase genes (IDH1 and IDH2) and the survival rates of patients. Isocitrate dehydrogenase-1 is an essential molecular biomarker, which can be present either in the wildtype or in the mutated state. [3] The survival rate of patients, whose tumors contained the mutated IDH gene, was significantly higher than that of patients with tumors containing wildtype IDH. [4] There are also mutation-specific treatment options available for IDH-mutated patients. [5] Unfortunately, the IDH mutation status can only be determined by genome sequencing after a tissue biopsy, which demands invasive brain surgery. [6]

The aim of this project is to train and test four different machine learning models, to find out if there is a way to determine the IDH1 mutation status solely based on non-invasive clinical and radiomic features. Radiomic features are numerical values, which describe shapes, colours and intensities of images. The radiomic features are extracted from MRI images, collected in the University of Pennsylvania glioblastoma (UPenn-GBM) data set. [7] This is currently the largest publicly available data set. Predicting the IDH mutation based on radiomic features has previously been attempted, but mostly based on smaller data sets and not utilising and comparing several different models. [2]

2 Methods

2.1 Data source

The provided data set was sourced from The Cancer Imaging Archive (TCIA). [8] Three files were supplied by the course administration, which differ slightly from the data sets from the TCIA.

- The provided file "clinFeatures_UPENN" is a preprocessed version of the TCIA file "UPENN-GBM_clinical_info_v2.1" downloadable under the title "Clinical Data" from the TCIA repository.
- The provided file "radFeatures_UPENN" is a merged data file of all 33 TCIA files saved with "automaticsegm" in the file name, saved in the zip folder "radiomic_features_CaPTk" downloadable under the title "Radiomic Data" from the TCIA repository.
- The provided file "UPENN-GBM_CaPTk_fe_params" is a copy of the TCIA file "fe_params_UPENN-GBM_CaPTk" available in the same aforementioned TCIA zip folder.

In the course of our project one additional file from the TCIA was utilized, the file "UPENN-GBM_CaPTk_radiomic_features_list" downloadable under the title "CaPTk radiomic features list" from the TCIA repository.

2.2 Preprocessing

2.2.1 Data Understanding and Cleaning

Understanding In the "01_Data Understanding (Radiological Features).ipynb" file the merged data set "radFeatures_UPENN.csv" was explored using the TCIA list saved in "UPENN-GBM_CaPTk_radiomic_features_list.csv" aiming to understand the radiomic features provided for our project.

In the "02_Data Preprocessing.py" file the missing and unique values of both the "radFeatures_UPENN.csv" and "clinFeatures_UPENN.csv" data sets were determined.

Further data understanding was achieved by visualisation, as mentioned in the section on data visualisation.

Cleaning Based on the number of missing and unique values certain columns were further inspected and based on the findings columns with suspicious values were dropped from the data set. From the clinical features data set all columns with only one unique value were eliminated and in the radiomic features data set all columns with up to three unique values were eliminated, since in most cases the three value entries there were NaN, zero and one random value, that seemed to be a default with no further informational benefit.

2.2.2 Feature and Label Selection

Features All radiomic features that were not previously dropped due to missing or faulty values were included in the feature data set. Additionally the clinical features "Age_at_scan_years" and "Gender" were also included, since they are the only two clinical features obtainable through non-invasive methods. Technically the Karnofsky performance score (KPS) is also available non-invasively but due to missing values and the subjectivity of the score this feature was excluded.

Label As a label the IDH1 mutation status was chosen. In addition to "wildtype" and "mutated", the column IDH1 included data labeled with "NOS/NEC". NOS indicates that there is insufficient or unavailable molecular information to make a specific diagnosis. [9] NEC denotes that analyses have been conducted, but the findings do not allow for a precise assignment to one of the categories within the WHO classification system, because they conflict with the WHO essential criteria. This is particularly relevant for "dual-genotype" IDH-mutant gliomas. [10]

By including the NOS/NEC cases, we attempted to enhance the model's robustness and generalizability, ensuring it can accurately predict glioblastoma IDH mutation status even in ambiguous or less clearly defined cases, thereby reflecting real-world clinical scenarios and improving its clinical applicability.

2.2.3 Data set splitting

The data set was split into a 80% training and 20% testing set to ensure the model accommodates for variance and to validate its performance on unseen data.

2.2.4 Missing value handling

Imputation To handle missing values an imputation method was applied. As a first attempt the IterativeImputer from scikit-learn [11] was run with default parameters, but the imputation turned out to be very computationally expensive. The imputation ran a long time and utilised a lot of RAM. 128 GB RAM on the Jupyter Euler Server were not enough and still threw a MemoryOutOfBounds error.

Estimator The default estimator was changed from the BaysianRidge to the RandomForestRegressor [12] to address this issue. The RandomForestRegressor however needs further hyperparameters. Unsupervised hyperparameter tuning was attempted by using GridSearchCV and setting y to None to prevent data leakage. According to the scikit-learn documentation, setting y to None should work [13], but an error message 'positional argument y_true missing' occurred, which we were unable to circumvent.

Based on the recommendation of a TA all default parameters were used, except for two hyperparameters to additionally target run time and memory usage. "n_estimators" and "max_iter" were both set to half the default value. No further hyperparameter tuning was attempted.

2.2.5 Feature selection

Given the high redundancy among radiomic features, it's crucial to reduce the number of correlated features to mitigate collinearity. The minimum redundancy maximum relevance (MRMR) feature selection method achieves this by evaluating a features correlation with the target variable and its redundancy with previously selected features in iterative steps, thereby maximizing relevance while minimizing redundancy. Previous studies employing MRMR in similar research contexts have demonstrated its effectiveness in improving model performance. [3] [14] Consequently, MRMR feature selection was utilized to select 300 features from the previous, cleaned up pool of 4457 features.

2.2.6 Synthetic oversampling

To deal with the class imbalance, SMOTETomek oversampling [15] was attempted to get a balanced data set. However, the models trained on the oversampled data seemed to be severely overfitting based on their high performance on the training set. Therefore, the model specific approaches described in the machine learning models section were preferred.

2.3 Data visualisation

Three plots were generated for the purpose of data understanding. A heat map visualises all missing value counts of the provided "radFeatures_UPENN.csv" data set, previous to any preprocessing.

The two other plots provide further insight into the chosen label. One plot is a countplot to visualise the distribution of label classes, the other is a histplot for the distribution of the clinical feature "Age_at_scan_years" with the label as a hue. The Age at diagnosis has been shown to have clinical relevance. [16]

2.4 Machine learning models

2.4.1 Random Forest Classifier *(Frank Julia)*

Model description The random forest classifier addresses multiclass classification by forming an ensemble of decision trees, each trained on random subsets of the data and features, and combining their outputs. This method introduces variability, reduces overfitting, and enhances robustness to outliers. [17] While random forests excel with high-dimensional data sets, their effectiveness decreases when many features are uninformative and vastly outnumber the samples, making feature selection necessary. [18]

Hyperparameters For the hyperparameter tuning, a custom scoring function was defined to calculate the mean of the macro F1-score, macro precision, macro recall, and accuracy. This was necessary to emphasize accurate predictions for the minority classes. The scoring function was then used with randomized search cross-validation, utilizing 3 folds and sampling 100 parameter settings. [19] Given that SMOTETomek is expected to cause less overfitting compared to SMOTE alone, hyperparameter tuning was conducted solely with the SMOTETomek data. [15] Eight hyperparameters were tuned to reduce overfitting and enhance model performance. Despite these efforts, predictions for the training data still indicated significant overfitting, as nearly all predictions were correct. Consequently, a different balancing approach was employed using the Balanced Random Forest Classifier from Imblearn. [20] The hyperparameter tuning process remained the same, with additional tuning for four more hyperparameters. The best parameter ranges were determined by additional research. [20] [21] [22] [23] [24]

2.4.2 XGBoost Classifier *(Buchser Fabienne)*

Model description The XGBoost classifier is an improved version of the Gradient Boosting Machine (GBM) algorithm. It is designed to be more scalable and accurate than its original. An XGBoost classifier builds an ensemble of decision trees and minimizes a specific loss function, which can be customized by the user, by gradually adding them. XGBoost classifiers can utilize regularization to prevent overfitting and support parallel processing. The classifier supports multiclass problems and is able to handle complex data sets, which makes it a potentially good fit for the provided data set.

Additional preprocessing Scaling is not strictly necessary for an XGBoost classifier model, but it can improve the performance if applied. Since the different radiomic features are on severely different scales and in differing ranges standardizing them to have zero mean and unit variance can be beneficial. Scaling was implemented with the scikit-learn StandardScaler. [25]

Encoding categorical labels is necessary with the XGBoost Classifier. The label classes were encoded using the scikit-learn LabelEncoder. [26]

In preparation for hyperparameter tuning the training set was further split into a training and validation set using an 80/20 split ratio, while stratifying on the training label class.

To handle the imbalance in the label classes, class weights were calculated based on the label training data set. These class weights are matched to the corresponding samples so that sample weights can be applied during model fitting.

Hyperparameters Hyperparameter tuning was achieved using the hyperopt library. The fmin function minimized the loss function of the model repetitively based on a parameter space, which defined ranges for the different hyperparameters. The model was fitted on the training set and evaluated on the validation set with different combinations of hyperparameter values. The best combination of hyperparameters (based on the performance on the validation set) was then saved and applied to the testing set. [27]

As a loss function the macro f1 score was chosen to ensure a good model performance over all label classes. [28]

2.4.3 Support Vector Machine (SVM) *(Schär Sabine)*

Model description Support vector machines differentiate between classes by identifying the best hyperplane that maximizes the distance between the nearest data points from opposite classes. [29] SVMs are effective in high dimensional spaces, even in cases where the number of dimensions is greater than the number of samples. [30] They can be robust towards a small number of outliers [28] and are memory efficient. [30] Support vector classifiers can be used for multiclass classification, which was the goal of this project. Therefore, a support vector classifier was implemented using the sklearn.svm.SVC function. [31]

Additional preprocessing The hyperplane separating the different classes will be heavily influenced by features with larger values. Therefore scaling makes sense before training an SVM. [32] The scikit-learn StandardScaler function [33] was used to scale the data.

The SVC was trained on the original imbalanced data. The class_weight parameter of the SVC function (class_weight='balanced') was utilised to deal with the label class imbalance.

Hyperparameters Hyperparameter tuning was performed three times, once for each of the three kernels linear, poly, and sigmoid. During hyperparameter tuning, RandomizedSearchCV was applied. Since the initial aim was to detect as many mutated cases as possible, a custom scoring function was used for these three tunings. Compared to the default scorer, the custom scoring function applies more weight to the recall, precision and f1 score of the class mutated.

The tuning was performed with 'C':[0, 1, 10] and 'gamma': [1, 0.1, 0.01, 0.001, 0.0001] and one of the three mentioned kernels. Based on the performances of the three resulting models on the training set the sigmoid kernel was chosen, because upon inspection of the performance on the training set it showed the lowest risk for overfitting.

2.4.4 Logistic Regression *(Nagendran Varsa)*

Model description The multinominal logistic regression is an extension of the sklearn.linear_model.LogisticRegression. Since our label has three different classes (wild-type, mutated, and NOS/NEC) and is thus a multiclass problem, this method was chosen over the original. It permits the modelling of more than two categories, which can be accomplished by estimating separate coefficients for each class and using the Softmax-function to calculate the probability rate each. [34] [35]

In this model, the label is converted into multiple dummy variables (1/0), with each dummy variable having its own binary logistic regression. If there are M categories, M-1 binary logistic regression models will be created. The two main reasons for choosing this model is that this model is comparable to the linear regression but provides easily interpretable diagnostic statistics and that, this model is more reliable to disruptions in the multivariate normality. [36]

Additional preprocessing Initially, the 2D array was converted to a 1D array using the target variables. The StandardScaler function was then used to scale the features of the training and test data sets, so that they all fell into a similar value range. Since the label values are categorical, they were encoded using the scikit-learn LabelEncoder to prepare them for the logistic regression. In addition, to address the label imbalance in the data set the class_weight parameter was set to balanced.

Hyperparameters For hyperparameter tuning, GridSearchCV with the default scorer was used. An iterator with the maximum number of iterations, parameters C, penalty, solver, and class_weight was added. Finally, using GridSearchCV, the set of parameters with the best fit was identified and the model was trained with multinominal logistic regression.

2.5 Performance evaluation

Correctly identifying all the different label classes is crucial for this project, since its aim is to reduce the need for tissue biopsies. The severe class imbalance however impairs the prediction of the minority classes, which is why Precision, Recall, and F1-Score are essential metrics as they ensure more accurate identification of a minority class. [28]

Since it is a multiclass classification problem, these metrics had to be calculated for each class individually. To provide an overall assessment, macro and weighted averages were included. The weighted average accounts for class imbalance by considering the proportion of each class, while the macro average treats all classes equally. [37]

Additionally, overall accuracy was measured, and the Area Under the ROC Curve was plotted to further assess model performance. Given the imbalanced nature of the label classes, a confusion matrix was plotted to identify areas where the model may be struggling. [28]

2.6 Feature Importance

Due to the selection of a non-linear kernel in the support vector machine's hyperparameter tuning, it wasn't possible to extract feature importance directly. However, feature importance for the Random Forest and XGBoost was visualized using SHAP barplots, while Logistic regression's feature importance was plotted using its coefficients.

3 Results

3.1 Data overview

3.1.1 Provided data sets

The provided data sets consist of data from 611 patients. The radiomic data set includes 4752 features and the clinical data set includes 9 additional features.

The radiomic data set consists of a combination of 33 different MRI sequences and 144 different extracted radiomic features, which make up the total of 4752 features.

3.1.2 Features

Radiomic In the radiomic features data set there are a maximum of 158 missing values per single feature. Over a total of 611 patients, this is 25.86% of missing data for the features in question.

Only 12.12% of radiomic features display no missing data and 63.64% of all radiomic features display over 10% missing values.

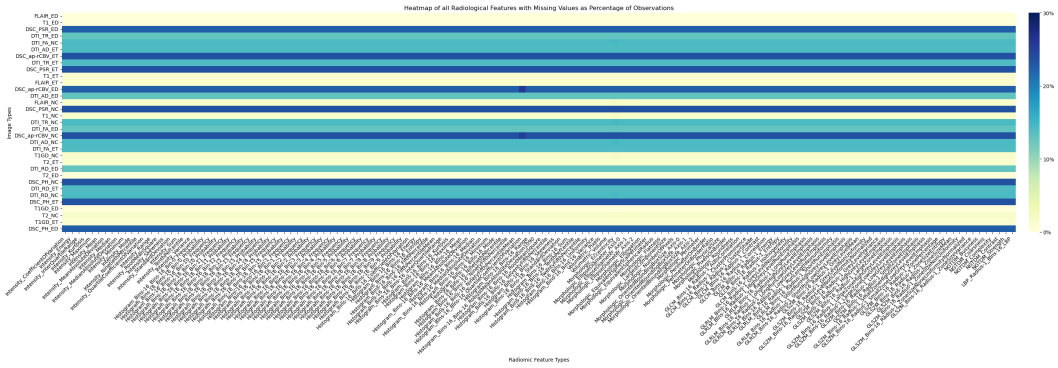
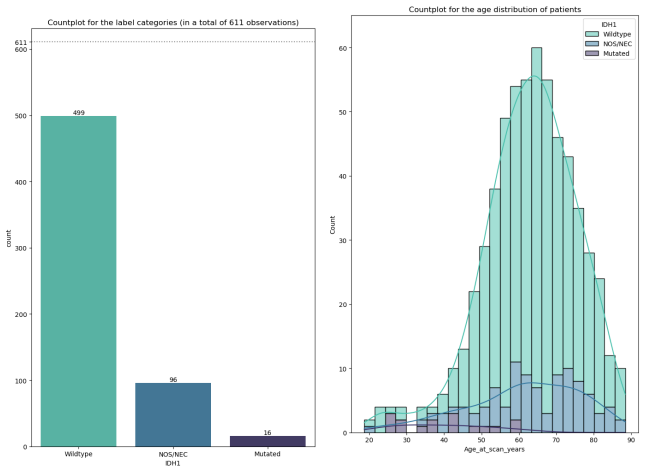


Figure 1: Radiomic Features, Missing Values Heat Map

Clinical The two included clinical features "Age_at_scan_years" and "Gender" both show no missing data.

3.1.3 Label

The label classes show an imbalance. The Wildtype is the majority class with 499 of 611 patients. The NOS/NEC and Mutated classes are the minority classes with 96 and 16 patients respectively.

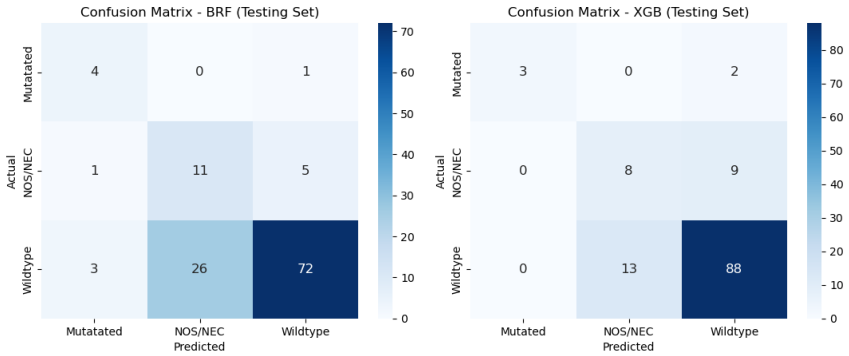


(a) Label distribution

(b) Age distribution with label as hue

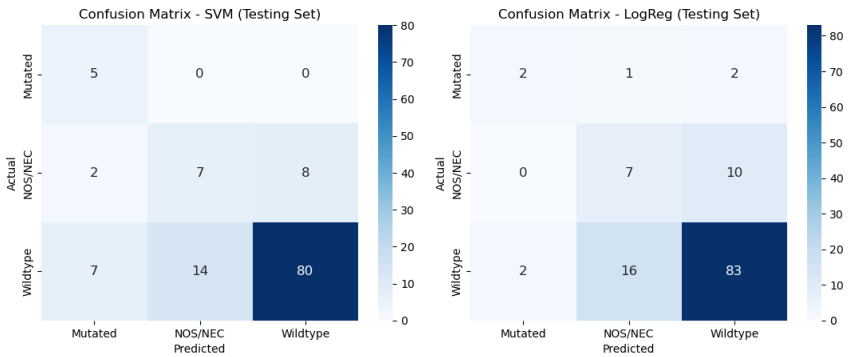
Figure 2: Histplots in regards to the target variable

3.2 Model performance



(a) Random Forest

(b) XGBoost



(c) Support Vector Machine

(d) Logistic Regression

Figure 3: Confusion Matrices for all models

Table 1: Precision (prec.), recall and F1-score (F1) for each class, as well as the macro and weighted average for all models; the support indicates the number of patients per class

	BRF			XGBoost			SVM			LogReg			Support
	prec.	recall	F1	prec.	recall	F1	prec.	recall	F1	prec.	recall	F1	
Mutated	0.50	0.80	0.62	1.00	0.60	0.75	0.36	1.00	0.53	0.50	0.40	0.44	5
NOS/NEC	0.30	0.65	0.41	0.38	0.47	0.42	0.33	0.41	0.37	0.29	0.41	0.34	17
Wildtype	0.92	0.71	0.80	0.89	0.87	0.88	0.91	0.79	0.85	0.87	0.82	0.85	101
macro avg	0.57	0.72	0.61	0.76	0.65	0.68	0.53	0.73	0.58	0.56	0.54	0.54	123
weighted avg	0.82	0.71	0.74	0.82	0.80	0.81	0.81	0.75	0.77	0.78	0.75	0.76	123

Table 2: Accuracy and ROC AUC for all models

	BRF	XGBoost	SVM	LogReg
ROC AUC (Mutated)	0.98	0.96	0.96	0.95
ROC AUC (NOS/NEC)	0.79	0.76	0.69	0.69
ROC AUC (Wildtype)	0.76	0.78	0.74	0.72
macro-avg ROC AUC	0.85	0.84	0.81	0.79
accuracy	0.71	0.80	0.75	0.75

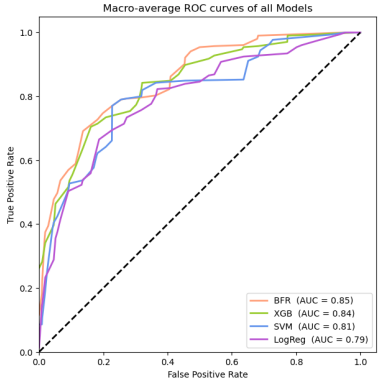
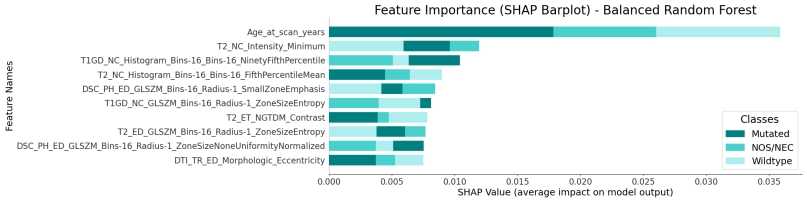
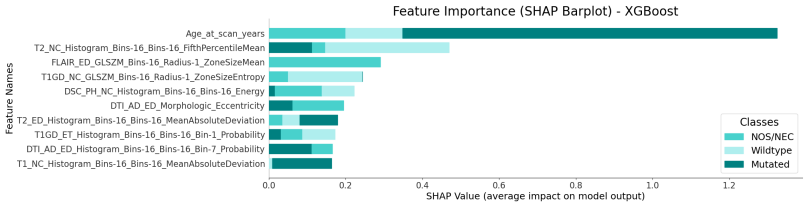


Figure 4: Macro-avg ROC curves

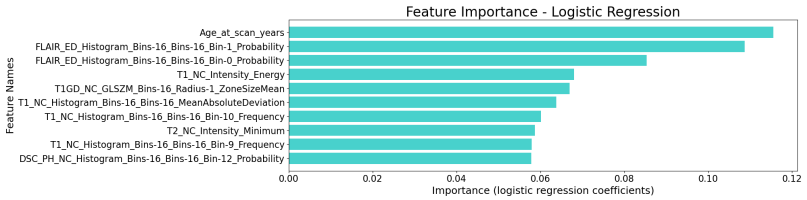
3.3 Feature Importance



(a) Top 10 features of the balanced random forest classifier



(b) Top 10 features of the XGBoost Classifier



(c) Top 10 features of the Logistic Regression

Figure 5: Feature Importance

4 Discussion

4.1 Interpretation

4.1.1 Model performance

Confusion matrix The Support Vector Machine accurately predicts all 5 mutated cases of the test set. Yet, given the small sample size, this may change for a different data or test set. Especially considering the marginal difference of one to three samples compared to other models. When targeting NOS/NEC predictions, the Balanced Random Forest classifier demonstrates superior performance in maximizing true positives. However, both the Support Vector Machine and Random Forest fall short compared to XGBoost in accurately identifying the wildtype and minimizing type 1 errors.

Therefore, each model has its advantages and drawbacks, with Logistic Regression showing comparatively weaker performance.

Precision, Recall and F1 score The performance metrics across all models indicate that the model predictions for NOS/NEC are very inaccurate. The scores for the mutated patients are somewhat better, though still not considered very good. Conversely, only the performance for the wildtype is quite good across all models. Given the poor performance for NOS/NEC, the macro average, which calculates the mean across all classes, is also on the lower side. In contrast, the weighted average is higher due to the strong model performance of the majority class and the reduced impact of the underperforming minority classes.

Comparing the models reveals that Logistic Regression demonstrates marginally worse average performance, while the XGBoost Classifier consistently outperforms the other models slightly. Therefore, if selecting a single model were necessary, XGBoost would be the preferred choice.

Accuracy and ROC AUC The area under the ROC curve slightly favors the Random Forest Classifier, but XGBoost exhibits the highest accuracy by a clear margin. The macro-average for the area under the ROC curve is quite similar for all models. In all models the ROC AUC curve for the mutated class was the best, when looking at the ROC AUC curves for each class separately. Sight the additional ROC Curve plots for further insights.

Performance in literature While our study has shown good results, other studies used different strategies to achieve a slightly better performance. For instance, they included only MRI sequencing techniques like T2 and FLAIR, which provide crucial details about peritumoral edema. Additionally, normalizing MRI image intensity, as suggested by some studies, could also play a role in improving predictive accuracy. In contrast to our data set other studies accessed multi-center data to keep their data set better balanced, avoid missing values and introduce more realistic variances. [3] [38]

4.1.2 Feature Importance

Across all models that could output feature importance, age consistently stood out as the most important feature. However, the importance of other parameters varied among models, with only a few overlaps.

The low prevalence of older patients with IDH1 mutations, which can also be observed in the plot depicting the age distribution of the used data set, explains this strong correlation. Other literature has also previously confirmed the same. Research suggests that the patients age at diagnosis can predict the IDH mutation status with a 90% accuracy. [16]

4.2 Limitations and points for improvement

The analysis of the data set showed that we had to deal with various problems.

Imputation The imputation of missing data was handled by using default parameters, which could have been optimized by tuning these hyperparameters with unsupervised methods. Other research papers show that they optimized their parameters for the highest ROC AUC score, because these are less affected by class distribution than accuracy. [2]

Feature selection Another point for improvement is our feature selection method MRMR, which focuses more on the relevance to the target variable and neglects the interdependence between the features. It could also not be ruled out that MRMR hasn't removed potentially important features. [39]

Label imbalance The class imbalance in our data set proved to be an other issue for model performance. The classes mutated and NOS/NEC have only few samples compared to the wildtype class. This makes predictions unreliable and models favour the majority class, wildtype.

Overfitting Several of the mentioned issues can additionally lead to overfitting, so targeting the underlying issues can also help reduce model overfitting.

4.3 Conclusions

This project found, that it is indeed possible to determine the IDH mutation status based on clinically non-invasive features. It is important to note that not all utilized models performed the same and that some of them like the XGBoost Classifier or the Balanced Random Forest model might be more fitting to the task than others. Nevertheless, in future studies it would be important to improve upon the challenges that where faced during this project.

Moving forward, the performance of our models could be improved by increasing the sample size, thus reducing imbalance, reducing missing value counts and analysing the clinical relevance of the predictive features, which are used to train the models. All of these improvements however would require a lot of resources and in-depth clinical knowledge on the topic of Glioblastoma and its treatment options.

Exploring ensemble methods or alternative, improved ways for hyperparameter tuning could also better predictive performance but it often requires tremendous computational resources.

This project explored an extensive data set of radiomic features from several MRI sequence types and while analyzing such data sets could potentially lead to further insights into which features and sequencing techniques show the most predictive relevance it also comes with challenges, that a smaller data set, which for example only focuses on one MRI sequence type, would not have to deal with. Predicting the IDH mutation status based on non-invasively obtained features seems to be a promising approach but further research is still needed before any model precise enough could have actual clinical application.

References

[1] H. Xu, X. Chen, Y. Sun, X. Hu, X. Zhang, Y. Wang, Q. Tang, Q. Zhu, K. Song, H. Chen *et al.*, "Comprehensive molecular characterization of long-term glioblastoma survivors," *Cancer Letters*, vol. 593, p. 216938, 2024.

[2] Y. Sakai, C. Yang, S. Kihira, N. Tsankova, F. Khan, A. Hormigo, A. Lai, T. Cloughesy, and K. Nael, "Mri radiomic features to predict idh1 mutation status in gliomas: a machine learning approach using gradient tree boosting," *International journal of molecular sciences*, vol. 21, no. 21, p. 8004, 2020.

[3] J. Zheng, H. Dong, M. Li, X. Lin, and C. Wang, "Prediction of idh1 gene mutation by a nomogram based on multiparametric and multiregional mr images," *Clinics*, vol. 78, p. 100238, 2023.

[4] B. C. Christensen, A. A. Smith, S. Zheng, D. C. Koestler, E. A. Houseman, C. J. Marsit, J. L. Wiemels, H. H. Nelson, M. R. Karagas, M. R. Wrensch *et al.*, "Dna methylation, isocitrate dehydrogenase mutation, and survival in glioma," *Journal of the National Cancer Institute*, vol. 103, no. 2, pp. 143–153, 2011.

[5] D. Ye, S. Ma, Y. Xiong, and K.-L. Guan, "R-2-hydroxyglutarate as the key effector of idh mutations promoting oncogenesis," *Cancer cell*, vol. 23, no. 3, pp. 274–276, 2013.

[6] D. N. Louis, A. Perry, G. Reifenberger, A. Von Deimling, D. Figarella-Branger, W. K. Cavenee, H. Ohgaki, O. D. Wiestler, P. Kleihues, and D. W. Ellison, "The 2016 world health organization classification of tumors of the central nervous system: a summary," *Acta neuropathologica*, vol. 131, pp. 803–820, 2016.

[7] S. Bakas, C. Sako, H. Akbari, M. Bilello, A. Sotiras, G. Shukla, J. D. Rudie, N. F. Santamaría, A. F. Kazerooni, S. Pati *et al.*, "The university of pennsylvania glioblastoma (upenn-gbm) cohort: advanced mri, clinical, genomics, & radiomics," *Scientific data*, vol. 9, no. 1, p. 453, 2022.

[8] S. Bakas, C. Sako, H. Akbari, M. Bilello, A. Sotiras, G. Shukla, J. D. Rudie, N. Flores Santamaria, A. Fathi Kazerooni, S. Pati, S. Rathore, E. Mamourian, S. M. Ha, W. Parker, J. Doshi, U. Baid, M. Bergman, Z. A. Binder, R. Verma, R. Lustig, A. Desai, S. Bagley, Z. Mourelatos, J. Morrissette, C. Watt, S. Brem, R. Wolf, M. P. Nasrallah, S. Mohan, D. M. O'Rourke, and C. Davatzikos, "Multi-parametric magnetic resonance imaging (mpMRI) scans for de novo glioblastoma (GBM) patients from the university of pennsylvania health system (UPENN-GBM)," accessed: 2024-6-16. [Online]. Available: https://www.cancerimagingarchive.net/collection/upenn-gbm/

[9] S. H. Torp, O. Solheim, and A. J. Skjulsvik, "The who 2021 classification of central nervous system tumours: a practical update on what neurosurgeons need to know—a minireview," *Acta Neurochirurgica*, vol. 164, no. 9, pp. 2453–2464, 2022.

[10] D. E. Reuss, "Updates on the who diagnosis of idh-mutant glioma," *Journal of Neuro-oncology*, vol. 162, no. 3, pp. 461–469, 2023.

[11] "Iterativeimputer," accessed: 2024-6-16. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.impute.IterativeImputer.html

[12] "Randomforestregressor," accessed: 2024-6-16. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

[13] "Gridsearchcv," accessed: 2024-6-16. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[14] M. H. Lee, J. Kim, S.-T. Kim, H.-M. Shin, H.-J. You, J. W. Choi, H. J. Seol, D.-H. Nam, J.-I. Lee, and D.-S. Kong, "Prediction of idh1 mutation status in glioblastoma using machine learning technique based on quantitative radiomic data," *World neurosurgery*, vol. 125, pp. e688–e696, 2019.

[15] "Smotetomek," accessed: 2024-6-16. [Online]. Available: https://imbalanced-learn.org/stable/references/generated/imblearn.combine.SMOTETomek.html

[16] S. Zheng, N. Rammohan, T. Sita, P. T. Teo, Y. Wu, M. Lesniak, S. Sachdev, and T. O. Thomas, "Gliopredictor: a deep learning model for identification of high-risk adult idh-mutant glioma towards adjuvant treatment planning," *Scientific reports*, vol. 14, no. 1, p. 2126, 2024.

[17] C. Jutzler, "Lecture foundations of data science," *Week 9: Classification II*, pp. 63–82, 2024.

[18] D. Ghosh and J. Cabrera, "Enriched random forest for high dimensional genomic data," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 19, no. 5, pp. 2817–2828, 2021.

[19] "Randomizedsearchcv," accessed: 2024-6-16. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html

[20] "Balancedrandomforestclassifier," accessed: 2024-6-16. [Online]. Available: https://imbalanced-learn.org/stable/references/generated/imblearn.ensemble.BalancedRandomForestClassifier.html

[21] C. Chen, A. Liaw, L. Breiman *et al.*, "Using random forest to learn imbalanced data," *University of California, Berkeley*, vol. 110, no. 1-12, p. 24, 2004.

[22] B. Mohtadi, "In depth: Parameter tuning for random forest," *Medium*, 2017, accessed: 2024-6-16. [Online]. Available: https://medium.com/all-things-ai/in-depth-parameter-tuning-for-random-forest-d67bb7e920d

[23] W. Koehrsen, "Hyperparameter tuning the random forest in python," *Towards Data Science*, 2018, accessed: 2024-6-16. [Online]. Available: https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74

[24] S. Saxena, "A beginner's guide to random forest hyperparameter tuning," *Analytics Vidhya*, 2023, accessed: 2024-6-16. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/03/beginners-guide-random-forest-hyperparameter-tuning/

[25] KT. Handling continuous features with XGBoost for regression problems. Accessed: 2024-6-16. [Online]. Available: https://kojitanaka.hashnode.dev/handling-continuous-features-with-xgboost-for-regression-problems

[26] X. Song. Two machine learning algorithms to predict: Xgboost, neural network with entity embedding. accessed: 2024-6-16. https://songxia-sophia.medium.com/two-machine-learning-algorithms-to-predict-xgboost-neural-network-with-entity-embedding-caac68717dea.

[27] T. R. D. HyperParameter tuning — hyperopt bayesian optimization for (xgboost and neural network). accessed: 2024-6-16. https://medium.com/analytics-vidhya/hyperparameter-tuning-hyperopt-bayesian-optimization-for-xgboost-and-neural-network-8aedf278a1c9.

[28] S. Brüningk, "Lecture foundations of data science," *Week 10: Classification III*, pp. 33, 46–81, 2024.

[29] "What are support vector machines (svms)?" accessed: 2024-6-16. [Online]. Available: https://www.ibm.com/topics/support-vector-machine

[30] "1.4. support vector machines," accessed: 2024-6-16. [Online]. Available: https://scikit-learn.org/stable/modules/svm.html

[31] "Svc," accessed: 2024-6-16. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[32] "Does svm need feature scaling or normalization?" accessed: 2024-6-16. [Online]. Available: https://forecastegy.com/posts/does-svm-need-feature-scaling-or-normalization/

[33] "Standardscaler," accessed: 2024-6-16. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[34] "LogisticRegression," accessed: 2024-6-16. [Online]. Available: https://scikit-learn/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[35] J. Brownlee, "Multinomial Logistic Regression With Python," Dec. 2020, accessed: 2024-6-16. [Online]. Available: https://machinelearningmastery.com/multinomial-logistic-regression-with-python/

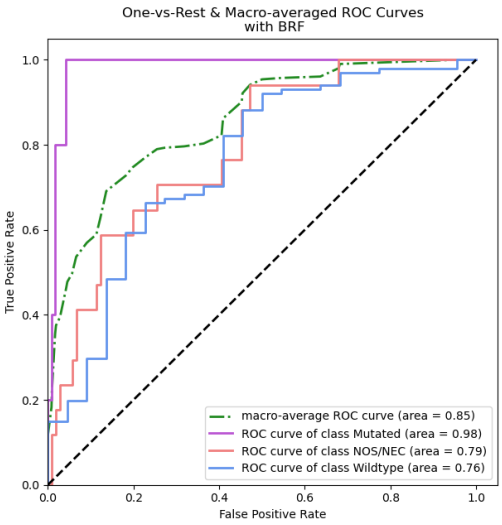
[36] A. Bayaga, "Multinomial logistic regression: Usage and application in risk analysis." *Journal of applied quantitative methods*, vol. 5, no. 2, 2010.

[37] Evidently AI Team, "Accuracy, precision, and recall in multi-class classification," accessed: 2024-6-16. [Online]. Available: https://www.evidentlyai.com/classification-metrics/multi-class-metrics

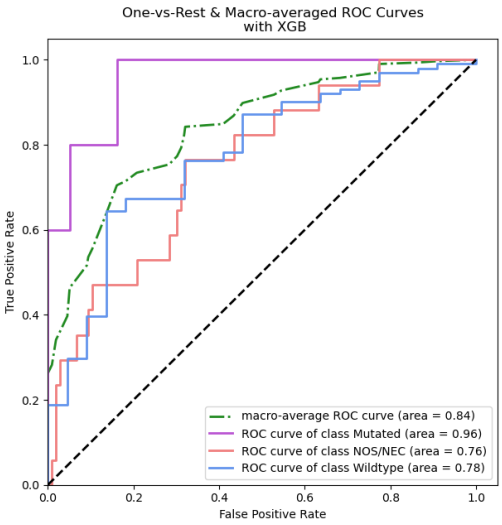
[38] H. Zhou, K. Chang, H. X. Bai, B. Xiao, C. Su, W. L. Bi, P. J. Zhang, J. T. Senders, M. Vallières, V. K. Kavouridis *et al.*, "Machine learning reveals multimodal mri patterns predictive of isocitrate dehydrogenase and 1p/19q status in diffuse low-and high-grade gliomas," *Journal of neuro-oncology*, vol. 142, pp. 299–307, 2019.

[39] T. Adesugba, "Mastering Feature Reduction: How mRMR Helps Machine Learning Models Cut Through the Noise," Apr. 2023, accessed: 2024-6-16. [Online]. Available: https://tobaml.hashnode.dev/mastering-feature-reduction-how-mrmr-helps-machine-learning-models-cut-through-the-noise

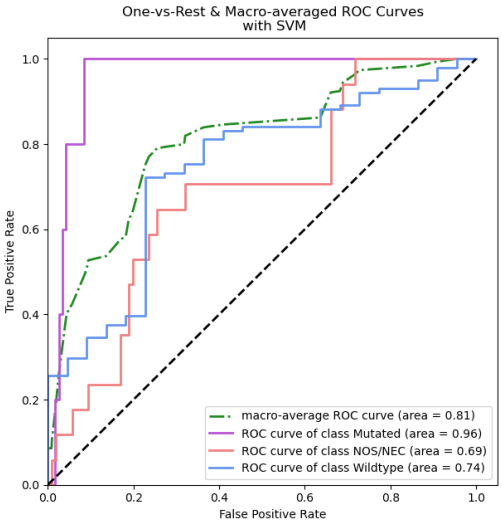
Additional ROC Curves (per model)



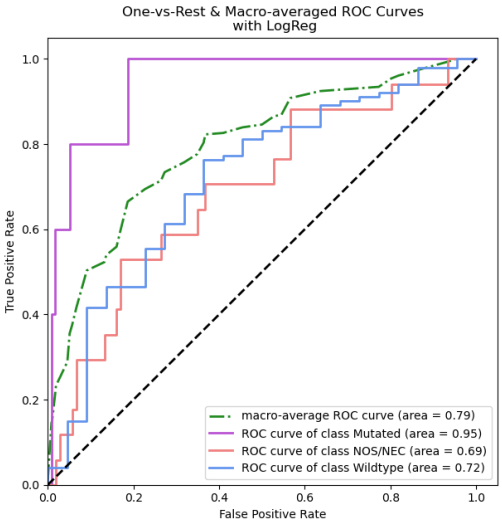
(a) Random Forest



(b) XGBoost



(c) Support Vector Machine



(d) Logistic Regression

Figure 6: ROC Curves for all models (depicting separate classes and macro-avg)

List of Figures

List of Tables

Python version and used packages

Name	Version
Python	3.11.8
hyperopt	0.2.7
imbalanced-ensemble	0.2.1
imbalanced-learn	0.12.2
jupyter	1.0.0
matplotlib	3.8.4
mrmr-selection	0.2.8
numpy	1.26.3
pandas	2.2.1
scikit-learn	1.4.2
seaborn	0.13.2
shap	0.42.1
xgboost	2.0.3