Enhancing Glioma Grading with Sparse, Interpretable Classification Models: A Comparative Investigation

Steven Tang

November 26, 2024

1 Background

1.1 Introduction

Gliomas are tumors arising from glial cells, which nourish, protect, and support neurons in the nervous system. There are two grades of glioma: lower-grade gliomas (LGG; grades I and II) and glioblastoma multiforme (GBM), a higher grade glioma. In oncology, whereas *stage* describes how much the tumor has spread, *grade* characterizes the aggressiveness of the tumor, so GBM is the most aggressive form. The grading of tumor cells involves a wide range of clinical, histological, and genetic features, and by characterizing the tumor cell's aggressiveness successfully, oncologists and other healthcare workers can effectively determine the quality and course of care, treatment, and prognosis for the patient. However, genetic testing for mutations of critical genes can be costly; therefore, it is best if we can find a small subset of those genes, coupled with demographic information, that can best characterize a glioma as GBM or LLG. In this study, we wish to compare various models exhibiting a wide range of sparsity/interpretability on one facet and predictive ability on the other.

1.2 Data Description

To investigate this problem, we use the "Glioma Grading Clinical and Mutations" dataset from the University of California, Irvine (UCI) Machine Learning Repository, derived from the Cancer Genome Atlas (TCGA). This dataset encompasses 3 clinical factors (including age, race, and gender) along with 20 binary variables characterizing whether the 20 genes associated with each variable are mutated.

There are a total of 839 observations of glioma patients originating from TCGA, each with 24 features (described in depth in Table 1) and no missing data. We first split the data so that 80% of it is used for training and the rest are held out for testing, as stratifying so that that class labels are proportionately split. Approximately 57.97% percent of the training data consist of LGG, whereas the other 42.03% consists of GBM, suggesting mild class imbalance.

Our first step was to plot frequencies of the non-genetic features of the dataset shown in Figure 1. This includes a histogram of the age at diagnosis (in years, with days in the year forming the decimal part of the data point), the only continuous feature that we have, as well as bar plots of race and gender. It appears that White patients seem to be disproportionately more represented than all other races, followed by Black and African Americans, Asians, and American Indians and Alaskan Natives. Additionally, there appears to be more male than female patients. The ages appear to be bimodal, with peaks around ages 35 and 65.

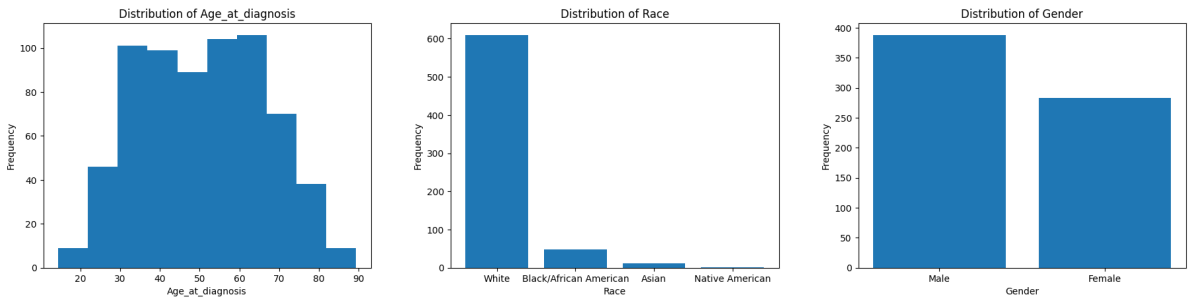
Since race is a nominal categorical variable rather than ordinal, we decided to one-hot encode each level; in other words, each level of race is given its own column, with entries indicating whether or not the patient is of that particular race. Due to there being a miniscule amount of American Indians and Alaskan Natives, we decide to remove the column; doing so also allows for the reduction of redundant information since zero values for each of these other binarized columns for race indicate that the patient is indeed American Indian and Native American. Since all of the features except the age at diagnosis are now binary, we decide to use min-max scaling on the age at diagnosis, so all variables are between 0 and 1.

We then look at the mutation counts of various genes in Figure 2. There appears to be as many mutated IDH1 and TP53 genes as non-mutated ones. All other genes suggest extreme level imbalance within the training dataset, though to a lesser extent for ATRX, PTEN, EGFR, CIC, and MUC16.

The relationships of the mutations of genes and whether or not a glioma is LGG or GBM is of immense interest before fitting any classification model. We sample 100 individuals from the dataset, in a manner that corresponds to the class imbalance; we select 58 of them to be lower-grade glioma patients and the other 42 to be glioblastoma multiforme patients. In Figure 3, we plot whether or not the given gene is mutated for the patient. There, we notice that LGG patients tend to have more instances of IDH1, TP53, CIC, and ATRX mutated compared to GBM patients, where GBM patients tend to have more PTEN and RB1 genes mutated. We then plot a receiver operator characteristic (ROC) curve in Figure 4 for each feature; each curve then represent the ROC for a predictive model (namely logistic model) involving only the given feature. Since almost all of the features are binary, notice that all but the Age curve consists of only two lines; on the other hand, the Age curve due to the number of unique values for that feature seems be smoother. Immediately, we see that age, as well as the mutations of IDH1, PTEN, and ATRX, among others, seem to be the most informative features to consider in a predictive model, aligning well with our observations about Figure 3.

[1]For brevity in this table, we use "black" for the level "black and african american" and "native" for "american indian or alaska native," as it appeared in the original data description

[2]All genetic binary variables are encoded as wild-type/normal (0) or mutated (1)



(a) Age at Diagnosis (b) Race (c) Gender

Figure 1: Histograms and barplots of non-genetic features in training data.

Figure 2: Barplots of Genetic Features in the Training Data

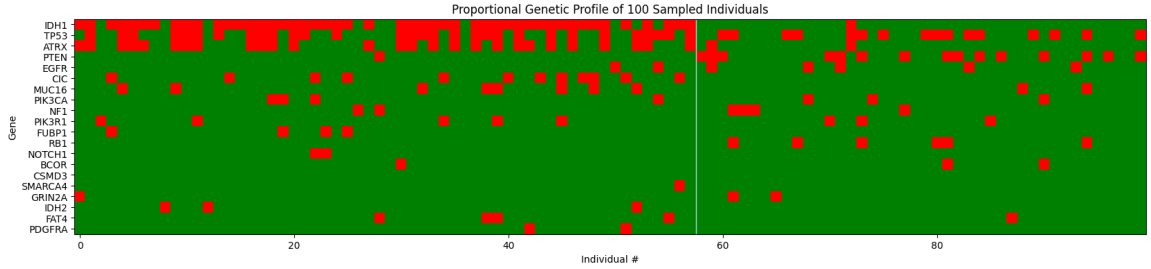


Figure 3: Genetic Profile of 100 individuals (58 LGG patients and 42 GBM patients). The LGG and GBM patients are separated by the faint blue line. Red indicates the mutation of the gene, while green indicates that the gene is normal.

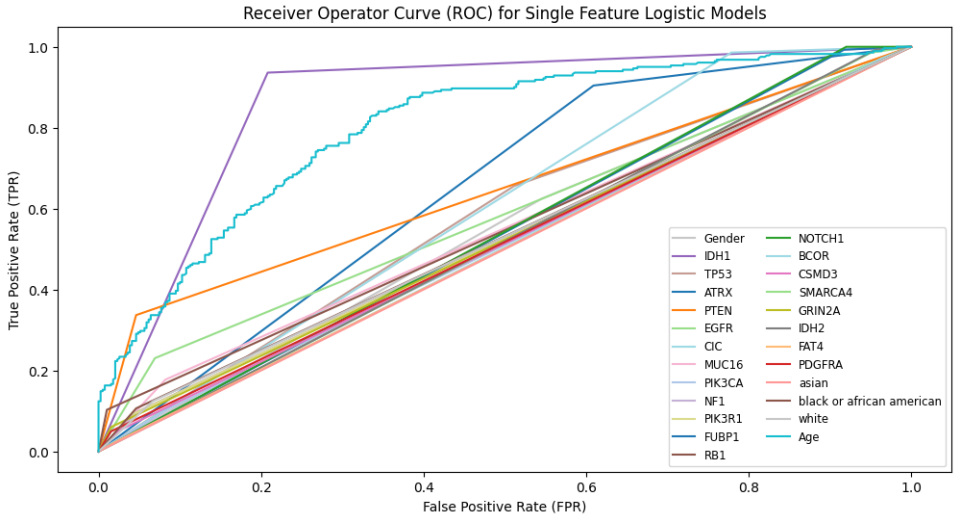


Figure 4: ROC Curves for (dummy) logistic regression models containing only the given feature.

2 Methods

To build a machine learning model for classifying whether observations associated with patients are lower-grade gliomas (LGG) or glioblastoma multiforme (GBM), we use logistic regression with least absolute shrinkage and selection operator (LASSO); decision trees; AdaBoost; Random Forests; and FastSparse generalized additive models (GAMs). We opt to use balanced accuracy as a metric for model comparison. We use Scikit-learn to fit L_1-penalized logistic regression, decision trees, AdaBoost, and random forest models, and the FastSparseGAMs package to fit FastSparse GAMs.

2.1 Classification Algorithms

2.1.1 Common (Scikit-Learn) Classification Algorithms

Logistic regression attempts to predict the log-odds ($\ln \frac{\mathbb{P}(Y=1)}{\mathbb{P}(Y=0)}$) from a linear combination of the features. We utilize LASSO so that we incorporate an L_1-penalty which regularizes the model, shrinking the coefficients to zero, so the model can perform feature selection. In doing so, we can end up with an interpretable, sparse model and reduce the potential of overfitting, doing so with low computational cost. However, logistic regression makes strong assumptions about the data, that the log-odds ratio is linear and that multicollinearity between predictors does not exist or is minimal.

Decision Trees are interpretable and intuitive, especially in the medical context due to the extensive use of boolean logic stemming from the decision rules that make up the tree. From the training data, the Classification and Regression Tree (CART) algorithm learns a series of decision rules (in tree form) that predict the target variable. The computational problem of finding the optimal decision tree is NP-complete, and the CART algorithm is

Table 1: Descriptions of the various variables in the TCGA dataset.

VARIABLE	DESCRIPTION	TYPE
Grade	Glioma Classification (0 = LGG; 1 = GBM)	Target
Gender	Gender (0 = Male; 1 = Female)	Binary
Age_at_diagnosis	Age at diagnosis (in years and days as decimal)	Continuous
Race	Race[1](0 = white; 1 = black; 2 = asian; 3 = native)	Categorical
IDH1	isocitrate dehydrogenase	Binary
TP53	tumor protein p53	(Genetic[2])
ATRX	ATRX chromatin remodeler	
PTEN	phosphotase and tensis homolog	
EGFR	epidermal growth factor receptor	
CIC	capicua transcriptional repressor	
MUC16	mucin 16, cell surface associated	
PIK3CA	phosphatidylinositol-4,5-bisphosphate 3-kinase catalytic subunit alpha	
NF1	neurofibromin 1	
PIK3R1	phosphoinositide-3-kinase regulatory subunit 1	
FUBP1	far upstream element binding protein 1	
RB1	RB transcriptional corepressor 1	
NOTCH1	notch receptor 1	
BCOR	BCL6 corepressor	
CSMD3	CUB and Sushi multiple domains 3	
SMARCA4	SWI/SNF related, matrix associated, actin dependent regulator of chromatin, subfamily a, member 4	
GRIN2A	glutamate ionotropic receptor NMDA type subunit 2A	
IDH2	isocitrate dehydrogenase (NADP(+)) 2	
FAT4	FAT atypical cadherin 4	
PDGFRA	platelet-derived growth factor receptor alpha	

greedy, only making locally optimal steps. In doing so, the decision tree found may not be the most optimal. Furthermore, the decisions tree could be unnecessarily complicated (having high depth to model simple concepts, like a diagonal boundary) due to there being rough transitions, and it is sensitive to small variations in data. Fortunately, decision trees allow for pruning to mitigate at least some of these ill-effects.

To mitigate the risk of overfitting and improve generalizability and robustness, we decide to investigate to ensemble methods, namely AdaBoost and random forests, a boosting and bagging algorithm, respectively.

AdaBoost combines a sequence of weak learner (each of which does slightly better than random guessing) in such a way that, any subsequent weak learner does better at what the previous ones miss. It does so by reweighting the data so that incorrectly predicted observation are weighted higher and weak learning algorithm is reapplied, and then placing weights on each of those weak learner. Unfortunately, AdaBoost can be sensitive to outliers (always focusing on difficult to predict observations), overfit if the base model is complex, potentially

focus more on the minority class when there is an imbalance in classes, be computationally expensive to perform, and lack interpretability. In using AdaBoost, we may trade off interpretability for predictive ability.

Random forests, on the other hand, incorporates two sources of randomness to improve generalizability. First, it takes a bootstrap sample of the training data, building independent trees in parallel on each. Second, when deciding on a feature to split, it chooses a random subset of a particular size. Additionally, though the combination of various decision trees through the averaging of predicted probabilities may obscure and remove any sense of interpretability, random forests do offer some notion of it with feature importance. However, random forests can be quite computationally expensive to fit. To limit the complexity of the trees that may be grown, we focus on 3 hyperparameters, namely the number of estimators (trees to grow), the maximum number of samples, and the maximum number of features to select.

In each of the previous classification algorithms, we use grid search with stratified 5-fold cross validation to tune the aforementioned hyperparameters (we describe the parameter setting).

2.1.2 Novel Classification Algorithms

Finally, **FastSparseGAMs** (Liu et al. 2022) fits sparse generalized additive models on the training data. Recall that GAMs consist of sums of a nonlinear function application on each feature, and it has a close connection with Adaboost, in that the decision stumps can be rearranged and added together according to their weights. The FastSparse algorithm works similarly, but does so with a different objective function (utilizing L_0-penalty, a penalty on the number of nonzero coefficients). It offers immense interpretability in the form of shape functions. The lack of continuous variables and abundence of binary variables is still suitable for use of FastSparseGAM, but the shape function that may arise may be dull. Additionally generalized additive models do not assume interactions between features, as opposed to decision trees which do. The algorithm requires binarizing the continuous "Age" feature.

Overall, we have a set of algorithms that fit models that differ in interpretability and predictability. With decision trees, logistic regression with LASSO, and FastSparseGAMs, interpretability is stressed more when compared to algorithms built to reduce overfitting and improve generalizability like AdaBoost and random forests.

2.2 Metrics

We opted in this investigation to use *balanced accuracy*:

$$\text{Balanced-Accuracy} = \frac{1}{2}(\text{Sensitivity} + \text{Specificity}) = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{FP+TN}\right)$$

In various healthcare contexts like our setting, specificity and sensitivity is commonly used. On the one hand, sensitivity asks, what percent of the GBM instances were correctly classi-

fied as GBM? This is the same as recall in classification tasks. On the other hand, specificity quantifies what the of LGG instances that were classified as LGG; this is the probabilistic complement of the false positive rate, i.e. of all of the LGG instances, how many of them were incorrectly predicted as being GBM. Both specificity and sensitivity is crucial as misclassification of an instance of either LGG or GBM may lead to poor prognoses or expensive care. For example, it is generally harder and more expensive to treat GBM than LGG, and GBM is more aggressive. Incorrect grading may thus lead to worsening of condition if a patient actually had GBM, or a treatment that may not be effective and is more expensive than necessary if the patient actually had LGG. Considering these questions, it makes sense we adopt the metric that combines these two measures into one unified metric.

3 Experiments

As stated previously, we performed a stratified split on the whole data, so 80% of it would be used for training and the other 20% for testing. Utilizing the best hyperparameter settings as described below, we refit the training data using and evaluated it on the held-out test set.

3.1 Results

Based on the balanced accuracy, random forest appears to perform the best (0.862), followed by logistic regression with LASSO and Decision Tree (0.853), and finally AdaBoost. We see that the model also very noticeably outperforms all others in their receiver-operator characteristic (ROC) curves and thereby their AUCs. Logistic regression with LASSO and decision trees give the greatest interpretability compared to the other model; in fact we easily display them in Table 3 and Figure 7. Meanwhile, random forests perform the best but it suffers in interpretability.

Table 2: The optimal hyperparameter setting and their associated balanced accuracy on the test set.

Classification Algorithm	**Hyperparameter Setting**	**Balanced Accuracy**
FastSparseGAM	$l_0 = 1.613$	0.862
Random Forest	max_depth = 8 max_features = 25 max_samples = 167 n_estimators = 128	0.862
Logistic Regression (LASSO)	$C = 1$ (default)	0.853
Decision Tree	ccp_alpha = 0.0001 max_depth = 4 max_leaf_nodes = 16	0.853
AdaBoost	learning_rate = 0.1 n_estimators = 512	0.846

However, based on the balance accuracy, FastSparse generalized additive models have the same balanced accuracy; however, the result should taken with caution since it was tuned less rigorously. Notwithstanding the lack of rigor in this result, it does demonstrate its competitiveness with other models, including random forests. More, as can be see in Figure 12, the model is highly interpretable compared to the others.

Table 3: The coefficients for logistic regression with LASSO model trained on all of the training data.

Feature	Coefficients	Feature	Coefficients
IDH1	-4.01	PDGFRA	0.0
IDH2	-2.63	PIK3CA	0.0
NOTCH1	-1.74	BCOR	0.0
NF1	-0.86	black...	0.28
EGFR	-0.60	RB1	0.33
CIC	-0.54	MUC16	0.38
SMARCA4	-0.29	CSMD3	0.58
Gender	-0.17	PTEN	0.69
asian	0.0	GRIN2A	0.77
ATRX	0.0	TP53	0.88
FAT4	0.0	PIK3R1	1.23
FUBP1	0.0	Age	2.03
white	0.0		

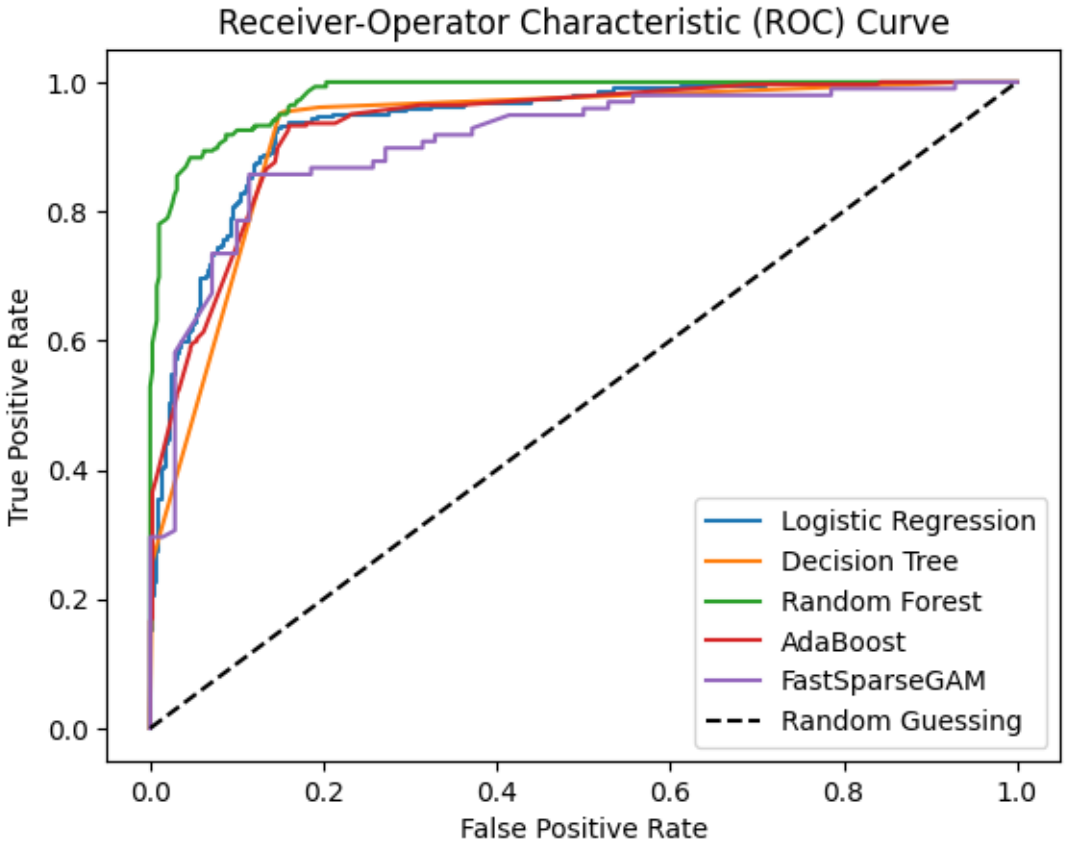


Figure 5: ROC Curves for all models considered.

3.2 Hyperparameter Selection

Table 4: The hyperparameters used for grid search for each classification algorithm.

Algorithm	**Hyperparameter**	**Values**
Logistic Regression	C	[0.01, 0.025, 0.05, 0.1, 0.5, 1]
Decision Tree	max_dept	[1, 2, 4, 8, 16]
	max_leaf_nodes	[2, 4, 8, 16, 32]
	ccp_alpha	[0.0001, 0.005, 0.01, 0.05, 0.1]
AdaBoost	learning_rate	[0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1]
	n_estimators	[1, 2, 4, 8, 16, 32, 64, 128, 256, 512]
Random Forest	max_features	[1, 5, 10, 15, 20, 25]
	max_samples	[335, 167, 83, 41] $((\lfloor \frac{n}{2} \rfloor, \lfloor \frac{n}{4} \rfloor, \lfloor \frac{n}{8} \rfloor, \lfloor \frac{n}{16} \rfloor))$
	n_estimators	[1, 2, 4, 8, 16, 32, 64, 128, 256]
	max_depth	[1, 2, 4, 8, 16]

Table 4 shows the hyperparameter grids we used to perform grid search for each classification algorithm. For each of the classification algorithms, we performed a grid search on hyperparameters (listed in the Table) utilizing stratified 5-fold cross validation to tune

the algorithms to find the hyperparameter combination that yielded the greatest balanced accuracy. In order to see how each hyperparameter affected the balanced accuracy score, we graph the score by a single hyperparameter while holding all other constant; again, this is in contrast to how grid search was used to find the optimal hyperparameter setting.

3.2.1 Logistic Regression with LASSO

For logistic regression, the regularization parameter C was tuned; more regularization is applied with smaller values of C. Regularization is achieved by putting greater weight on the L_1-penalty term. In Figure 6, we see that the less regularization there is, the greater the average balanced accuracy on folds of the training set.

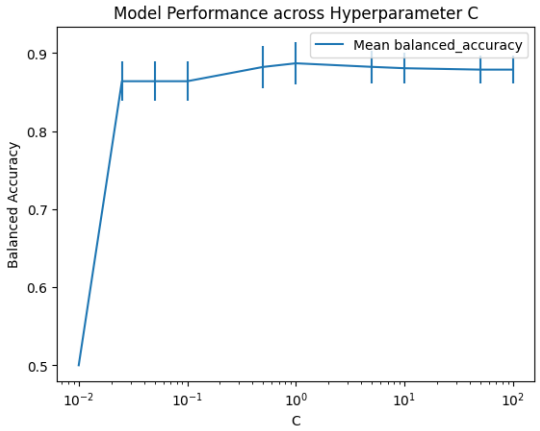


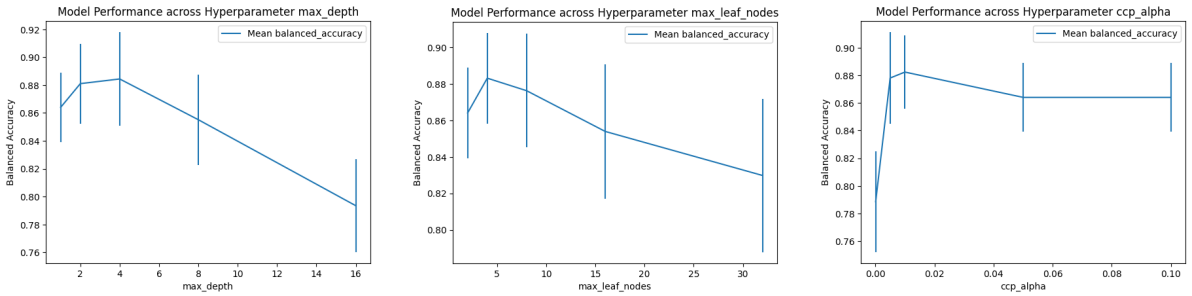
Figure 6: How model performance changes with $\log C$

3.2.2 Decision Tree

Three parameters are tuned for the CART algorithm to build decision trees according to the training data. Sparsity and potential for overfitting in a decision tree can depend on the depth of the tree and number of leaves, among other hyperparameters. We therefore decide to tune the maximum depth, maximum number of leaf nodes, and the cost-complexity pruning parameter α. Holding all other hyperparameters constant, performance seems to drop off after a maximum depth of 4, maximum number of leaf nodes of 5, and cost-complexity pruning parameter of approximately 0.01, as shown in Figure 7. The maximum depth and cost-complexity pruning parameter seem to be effect at improving the model's balanced accuracy.

3.2.3 AdaBoost

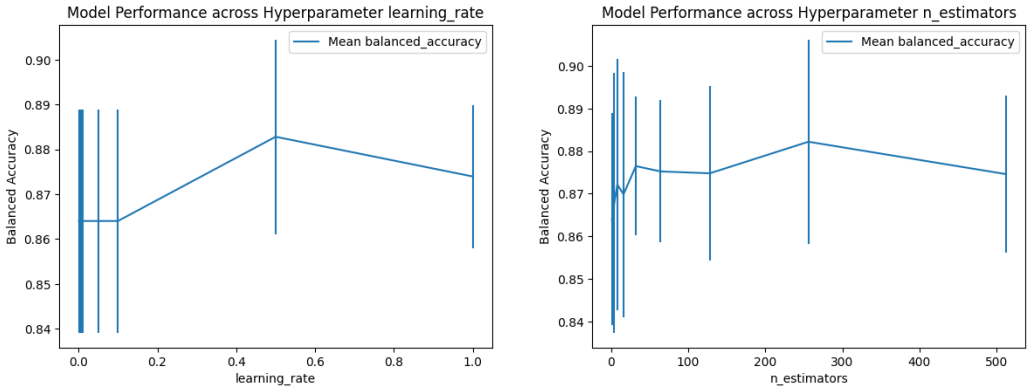
For the AdaBoost algorithm, we consider a weak learner that is a decision tree of maximum depth 1 (a stump), as well as hyperparameters such as the number of estimators and the learning rate. As shown in Figure 8, performance peaks around a learning rate of 0.5 and



(a) Maximum depth (b) Maximum number of leaf nodes (c) Cost-Complexity Pruning Parameter

Figure 7: How model performance changes with various decision tree hyperparameters (holding every other hyperparameter constant).

the utilization of 256 estimators. These dynamics are all within a single standard deviation of one another.

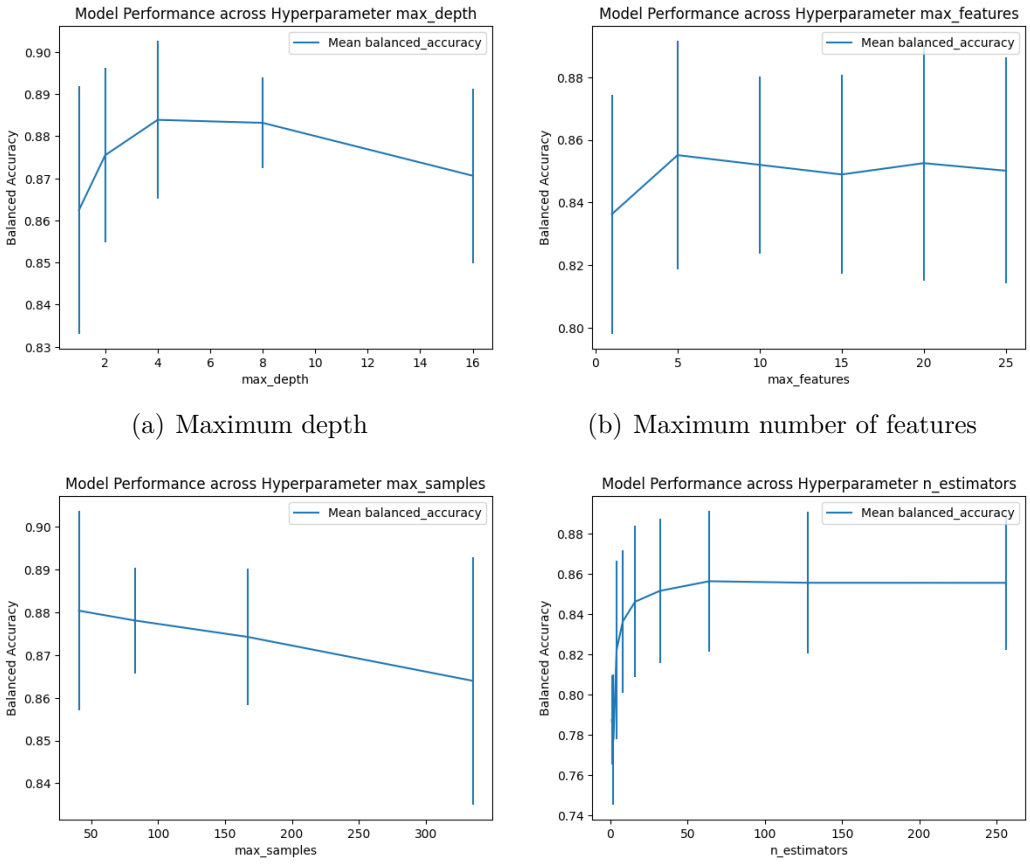


(a) Maximum depth (b) Maximum number of leaf nodes

Figure 8: How model performance changes with various AdaBoost hyperparameters (holding every other hyperparameter constant).

3.2.4 Random Forest

For the random forest algorithm, we consider four hyperparameter, common to both decision trees and the AdaBoost algorithm. Model performance, according to Figure 9 appears to drop off after a max depth of 4; stay constant over the maximum number of features, though maximized around 5 features; slightly decreases with maximum number of bootstrap samples. These dynamics are all within a single standard deviation of one another.



(a) Maximum depth

(b) Maximum number of features

(c) Maximum number of samples

(d) Number of Estimators

Figure 9: How model performance changes with various decision tree hyperparameters (holding every other hyperparameter constant).

3.2.5 FastSparseGAMs

For the FastSparseGAM algorithm, we consider a single hyperparameter for the regularization of the ℓ_0-norm term. Model performance, according to Figure 10 appears to stay fairly constant across $\log \lambda_0$ but it drops to 0.5 for large values.

3.3 Variable Importance Analysis

Below we plot the model reliance of each model; in other words, we permute each feature and plot the change in model score from the permutated and original datset. This measure gives how important a feature is to a the given model. We do so with the model trained on the whole data. Notice the great similarity between each of the models; namely it appears that IDH1, IDH2, NOTCH1, and Age all appear to be critical in predicting whether a tumor should be graded as GBM or LGG.

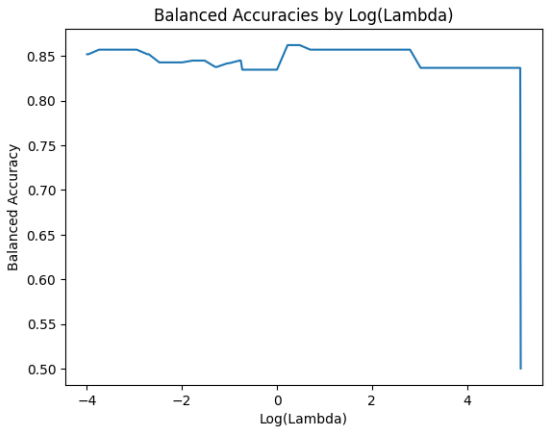


Figure 10: How model performance changes with $\log \ell_0$

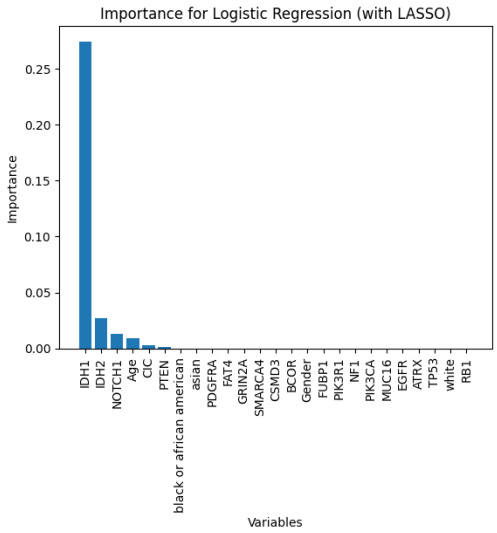
4 Insight

The interpretability of the models investigated in this report aligns well with exploratory data analysis and established biological findings regarding key markers that differentiate lower-grade gliomas (LGG) from glioblastoma multiforme (GBM). Among these, the mutation of IDH1 consistently emerges as the most significant marker across all models, exhibiting high variable importance scores. Additionally, age is identified as a critical factor, consistent with scientific literature indicating that GBM risk increases with age. These findings are robust across the range of models studied.

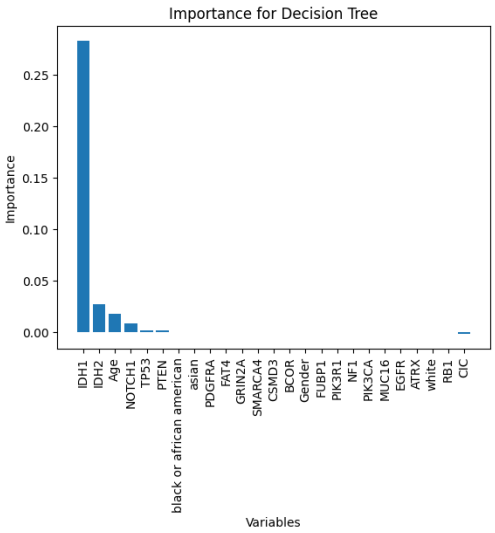
Despite these promising results, several issues persist with the TCGA glioma dataset. Notably, there is a lack of sufficient representation for certain feature levels. For example, the severe underrepresentation of American Indian and Alaska Native groups in the dataset is concerning. This imbalance complicates the ability of models to generalize and produce accurate predictions for these populations. This challenge is particularly pressing in healthcare, where underserved communities often experience disparities in treatment quality. Addressing the issue of small level sizes is essential for building models capable of making equitable and meaningful predictions.

Further improvements can be achieved by customizing loss functions to reflect real-world constraints and priorities. Conventional classification loss functions aim to reduce sparsity and misclassification errors but often neglect practical considerations such as cost and accessibility. For instance, as noted by Tasci et al., the IDH1 genetic test can be expensive and time-consuming. Incorporating data on the cost and efficiency of genetic tests into loss functions could lead to the identification of cost-effective combinations of tests with high predictive power. Expanding the dataset to include additional clinical features, such as histological information about tumor cell structure, may also enhance model performance.

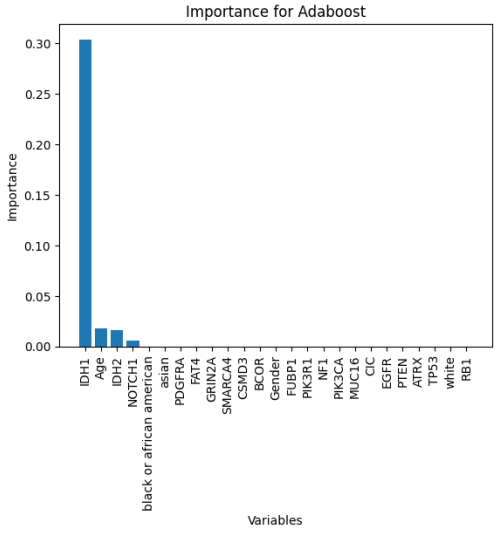
The pursuit of interpretable yet optimal models remains an important focus. Techniques like generalized optimal sparse decision trees (GOSDT), which utilize bounds and dynamic programming to construct optimal and sparse trees, offer a promising avenue. Such models retain the interpretability of traditional decision trees while improving predictive performance



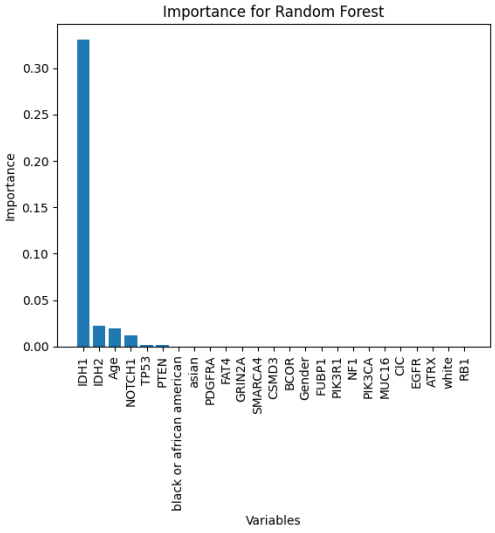
(a) Logistic Regression (With LASSO)



(b) Decision Tree



(c) AdaBoost

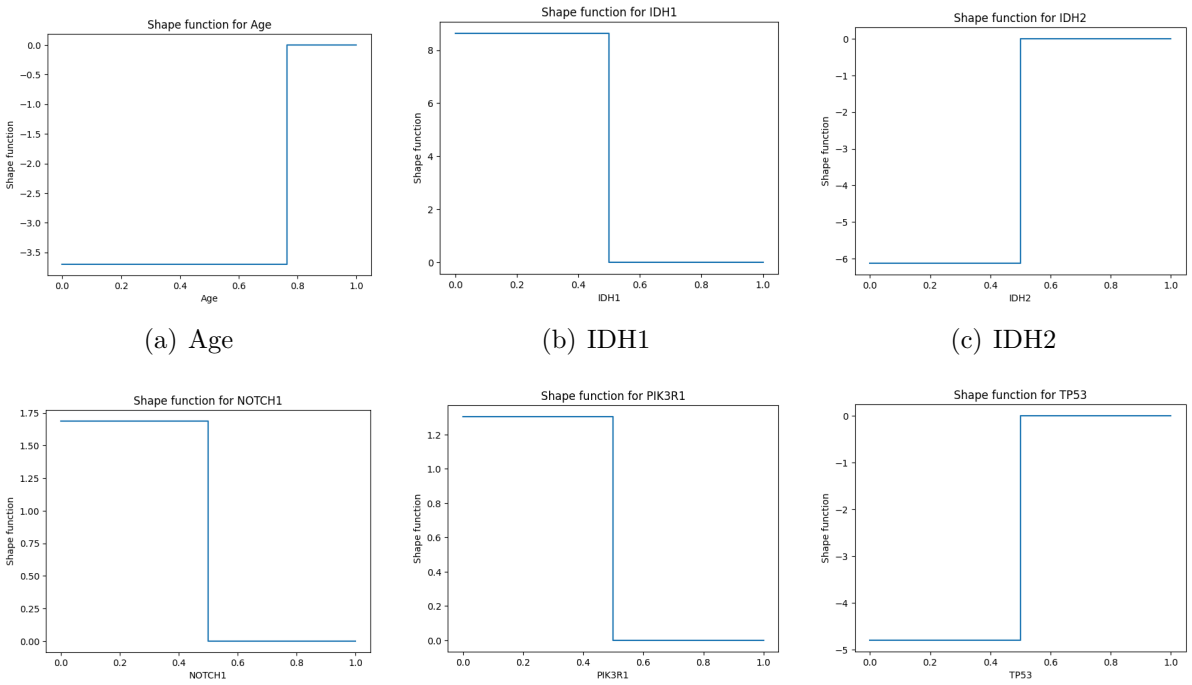


(d) Random Forest

Figure 11: Variable Importances (Model Reliance) for each model.

and reducing complexity. Investigating similar sparse and interpretable models tailored to this domain may yield significant advancements.

Effective glioma grading hinges on the selection of relevant genetic markers, which can improve patient care and facilitate targeted drug and treatment discovery. Sparse and interpretable models empower healthcare providers to conduct tumor grading efficiently and cost-effectively, reducing the burden on patients. Addressing dataset limitations and enhancing model design with real-world considerations will further strengthen the quality of care for glioma patients and offer valuable insights into the biology of these tumors.



(a) Age (b) IDH1 (c) IDH2

(d) NOTCH1 (e) PIK3R1 (f) TP53

Figure 12: Shape Functions for FastSparseGAM fit on training data, chosen to maximize balanced accuracy on test data.

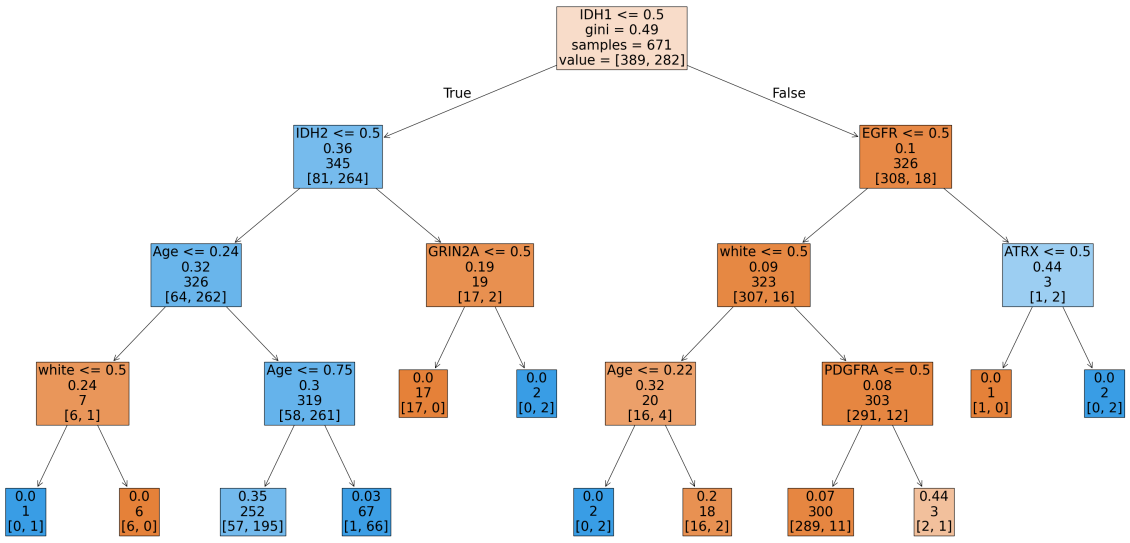


Figure 13: Decision tree trained on all of the training data

Citations

Liu, J., Zhong, C., Seltzer, M., & Rudin, C. (2022). Fast sparse classification for generalized linear and additive models. *International Conference on Artificial Intelligence and*

Statistics, 9304–9333. PMLR.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

Tasci, E., Camphausen, K., Krauze, A., & Zhuge, Y. (2022). Glioma Grading Clinical and Mutation Features [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C5R62J.