Integrated Predictive Modeling for Brain Tumor Outcomes

Aryan Mukherjee
University High School, Irvine, CA

Abstract

Brain tumors, most commonly gliomas and meningiomas, are abnormal masses of cells growing at the brain. We employ a Random Survival Forest model and an XG Boosting model, two ensemble types, to predict brain tumor patient survival from clinical data. XGBoost outperformed RSF with C-indices of 0.7143 and 0.6928, respectively, and Cramér's ϕ values of 0.5270 and 0.4188, respectively. Both models identified diagnosis as the most important feature and stereotactic methods as the least important. They also outperformed other prominent models tested on datasets with similar cardinality, and their accuracy can be improved with a larger sample size and the inclusion of omic data in training. The results are applicable in future ensemble model design and oncology survival forecasting.

Keywords: Brain tumor, ensemble learning, Random Survival Forest, Gradient Boosted Decision Trees, survival analysis

1 Introduction

1.1 Preliminary Information

Brain and other central nervous system (CNS) tumors are responsible for a significant health burden in the United States. Between 2016 and 2020, malignant CNS tumors caused approximately 17,206 deaths per year, corresponding to an annual mortality rate of 4.42 per 100,000 population. [1] Gliomas represent around 26% of all brain tumors, with glioblastoma being the most common malignant subtype; meningiomas are the most

prevalent non-malignant tumor. [1] These figures underscore the urgent need to identify prognostic factors that influence survival.

Survival analysis is the statistical framework for modeling time-to-event data in the presence of censoring. Censoring occurs when an individual who has not experienced an event yet leaves the study, or when the study concludes before an event is observed for that individual.

This paper is part three of a three-part series on predictive modeling for brain tumor survival outcomes. It focuses on two non-parametric approaches: the Random Survival Forest (RSF) model for survival time prediction, and a variation of the Gradient Boosting model. The second paper in this series, like this one, concerns the field of survival analysis, but it focused instead on parametric models such as Cox and log-logistic. [2] The previous paper also investigated the RSF model; however, a key difference between the Random Survival Forest in this paper and that of the previous paper is its purpose; this study's RSF is designed to predict survival time, while the previous study's RSF was designed to predict survival function.

The structure of this paper includes a data description, a brief literature review, methodological details, model-specific results, and implications for oncology and model selection.

1.2 Data Description

The study analyzes 87 patients treated from 2004 to 2011 at the Masaryk Memorial Cancer Institute in Brno, all of whom received radiation therapy [3]. The original dataset contained 88 patients, but we removed one due to a missing value. Of the remaining 87 patients, 52 observations were censored. Here is the complete list of variables:

- **Gender:** Patient's self-reported gender (male or female).
- **Diagnosis:** Type of brain tumor (e.g., glioma, meningioma).
- **Location:** Tumor location in the brain, categorized as infratentorial (below the tentorium cerebelli, affecting the cerebellum or brainstem) or supratentorial (above the tentorium, affecting the cerebral hemispheres).
- **Stereotactic Method:** Type of radiation therapy applied—SRS (stereotactic radiosurgery, single high-dose) or SRT (stereotactic radiotherapy, multiple lower-dose fractions).
- **Karnofsky Index (KI):** A performance status score ranging from 0 to 100, with higher values indicating better ability to carry out daily activities.

• **Gross Tumor Volume (GTV):** The measured physical volume of the tumor (in cm^3), derived from imaging.

• **Duration:** Survival time, measured in months from study enrollment to death (or last follow-up if censored).

• **Censoring:** Patient status at last observation: $\delta = 0$ indicates censoring (patient alive or lost to follow-up at last contact), while $\delta = 1$ indicates event occurrence (death).

1.3 Literature Review

Much literature has been produced in recent years on the topic of ensemble learning, and several authors have used machine learning models to analyze gliomas, particularly glioblastomas.

For instance, Kim et al. used a Random Forest regressor to predict the survival time of brain tumor patients from MR images rather than a survival-based dataset, resulting in a 50.5% average accuracy. [4]

Qiu et al. used a Random Forest regressor to predict progression-free survival time, a metric characterizing either time until death or time until tumor growth or spread. Eighty-two consecutive patients at a hospital in Shanghai were included in the study, and the Random Survival Forest produced a C-index of 0.611 on a test set. All patients had high-grade glioma; however, the mutation in the isocitrate dehydrogenase gene varied. This mutation, alongside age, was among the only features with positive feature importance, with age being the most important. The Karnofsky index, gender, and tumor location were not identified as important features [5].

Audureau et al. determined the feature importance of prognostic covariates in a dataset of 777 patients with supratentorial glioblastoma. Using two RSF models with differing computational costs and split mechanisms, the Karnofsky index was identified as the most important feature for training their RSF [6].

Karami et al. leveraged clinical data from 29 glioblastoma multiforme patients, none of whom were censored. Using Gradient Boosting decision trees to classify survival on the dataset, an accuracy of 75% was achieved using RSF-important features, while an accuracy of 58% was achieved using all features. Note that AUC was used, which binarizes survival and reduces model complexity [7].

Rajput et al. compared a Random Forest regressor and a Gradient Boosting regressor to predict survival on a very large dataset from 2020, exclusively made available for a brain tumor challenge. On the validation set, Random Forest and Gradient Boosting yielded

accuracies of 51.7% and 62.1%, respectively, giving Gradient Boosting the advantage. [8]

Charlton et al. conducted a binary analysis (split at the 1-year mark) of survival on 1283 patients, including censored data. Several machine learning techniques were used, including the two blackbox classifiers (i.e., classifiers with mechanisms that are difficult to precisely characterize) of Random Survival Forest and Support Vector Machine. Random Survival Forest yielded the highest accuracy out of all models, and identified age, diagnosis, and Karnofsky index as the three critical features. The rest of the models disagreed on the placement of age; however, they uniformly ranked diagnosis as the highest or second-highest feature and Karnofsky index near the middle. [9]

Renugadevi et al. performed survival prediction using ensemble and standard regression algorithms, incorporating an Extreme Gradient Boost model for its non-linearity. Their XGBoost model demonstrated the greatest accuracy, as well as the only degree of accuracy useful for accurate survival prediction ($R^2 > 0.7$). In terms of minimizing error in predictions, Random Forest was the next best model. [10]

Hachimy et al. used clinical data and omic data - data pertaining to the molecular and genomic makeup of patients - to perform machine learning regression on 619 glioblastoma patients with 12 unique features. The inclusion of omic data resulted in a significant increase in the values of C-indices compared to previous literature, producing a C-index of 0.78 and 0.80 for Random Survival Forest and Extreme Gradient Boosting, respectively. [11]

2 Ensemble Learning

One machine learning technique that enhances predictive power and has been popularized in the last decade is ensemble learning. Often utilized in regression models, as we will do here, and in neural networks, ensemble learning comprises several models that are constructed as layers that work together to predict a desired sample statistic. By aggregating the results of these models and building from the base function onward, ensemble learning models are capable of producing significantly more accurate predictions than the sum of their parts.

We will investigate two such ensemble learning techniques, namely "bagging" and "boosting," via Random Survival Forest (RSF) and Gradient Boosting, respectively. In addition to predicting the survival function, the RSF can predict the time until death for patients in the testing set, which we explore in this section. The algorithms used to run these models both optimize the bias–variance trade-off and minimize overfitting to the

training data, although the means through which they do so are different. Additionally, boosting models are run sequentially, while bagging models can be run in parallel, resulting in differences in model run time, which are relevant on an industrial scale but negligible for our small sample size. We will continue, as in the previous paper, to use proxy measures to compare the accuracy of these techniques.

2.1 Random Survival Forest to Predict Patient Survival Time

Random Forest is a non-parametric ensemble learning method useful for event prediction. It relies upon a set of decision trees, bootstrapped from the original data source, that are created from a select number of features (covariates) and used to predict a particular outcome. In the context of machine learning, Random Survival Forest (RSF) is a type of Random Forest that predicts patient survival time in the setting of clinical trials.

Traditionally, RSF has been used to estimate the survival function, due to its flexibility under high-dimensional data and complex relationships that exist between predictors. However, its architecture can also be manipulated to forecast patient survival time. As such, the prediction error curve for the RSF model, tracking the expected squared residual of the predicted survival function (weighted to account for censoring) over time, and the resulting integrated Brier score are omitted from our accuracy assessment, and alternate methods must be used when analyzing results.

Finally, given that our data include 52 censored and 35 uncensored observations, balancing the RSF is unlikely to affect model accuracy. Therefore, adjustment for class imbalance is unnecessary. Accordingly, we proceed with a standard RSF regression model. [12].

2.1.1 Methodology

A decision tree is a recursive partitioning structure that models event outcomes based on the likelihoods provided by a subject's features. In RSF, features are split based on favorability (goodness-of-split) of the child nodes, determined by the pair's capacity to yield the most significant log-rank test statistic upon comparison. That is, if the dataset is split by, for instance, a parent node of $X_1 > 45$ (where X_1 denotes a covariate), then that would mean that patients with $X_1 > 45$ have significantly different survival functions compared to those with $X_1 < 45$ at a significance level lower than for $X_1 > k$, where k is any constant from the set of all possible covariate values not equal to 45.

To reduce computation time, RSF employs randomized splitting, in which a random subset of candidate split points is selected at each node, and the best split among these is

identified using the log-rank test. For instance, rather than testing all possible integers $k \in (0, 100)$ for the Karnofsky index, the algorithm may test only a few randomly chosen values, such as $k \in \{5, 10, \ldots, 90, 95\}$, to determine which split produces the largest separation between survival functions of the two child nodes.

This feature-selection process can also be viewed as an operation that maximizes the reduction in entropy, or equivalently, maximizes the distinction between groups with differing risk profiles. Because the log-rank test accounts for right-censored data by including censored patients in the risk set up to the time of censoring, this feature-selection procedure is inherently compatible with right-censored observations and is therefore suitable for our dataset.

The steps to construct the RSF model can be summarized as follows (in the context of our sample):

Bootstrapping

Randomly select, with replacement, 87 rows from the dataset to form a bootstrapped dataset (B-set). Then, select $\sqrt{|X|} \approx 2$ unique features from $X = \{x_1, \ldots, x_6\}$ to construct the corresponding tree.

Repeat this procedure to generate 100 B-sets in total, as computational resources allow.

Validating

A decision tree is grown for each B-set using the log-rank splitting criterion, based only on the randomly selected features. Because sampling is with replacement, on average, about 36.6% of the original data are excluded from any given tree, forming the out-of-bag (OOB) set.

To see this, let n be the number of observations in a B-set. The probability that a given observation is not selected in one draw is $1 - 1/n$, so the probability that it is excluded from all n draws is $(1 - 1/n)^n \approx e^{-1} \approx 0.366$. Consequently, each observation appears in roughly 63.4% of the trees, with the remainder serving as OOB data.

For each patient, predictions are aggregated across the trees in which they are OOB to produce an unbiased estimate of survival time. This procedure can be applied to all observations in the dataset, effectively providing a validation set of the same size as the training set without overlap.

Evaluating

The Harrell's C-index is a widely used metric for assessing time-to-event predictions. Here, we describe its computation in the context of our patient data.

First, all $\binom{87}{2} = 3741$ ordered pairs (T_j, T_k) of observed survival times are formed, with $T_k \geq T_j$. Pairs where $\delta_j = 0$ are discarded; for tied pairs, if $\delta_j = \delta_k$, the pair is also excluded. For non-tied pairs where $\delta_k = 0$, the pair is retained, as the model can still be evaluated by comparing the predicted order $(\hat{T}_j, \hat{T}_k)$ to the observed order (T_j, T_k).

Let Z_i denote the concordance for the ith pair:

$$Z_i = \begin{cases} 1, & \text{if } (\hat{T}_j < \hat{T}_k \text{ and } T_j < T_k) \text{ or } (\hat{T}_j = \hat{T}_k \text{ and } T_j = T_k), \\ 0.5, & \text{if } (\hat{T}_j = \hat{T}_k \text{ and } T_j < T_k) \text{ or } (\hat{T}_j < \hat{T}_k \text{ and } T_j = T_k), \\ 0, & \text{otherwise.} \end{cases}$$

The overall concordance and C-index are then

$$\text{Concordance} = \sum_{i=1}^{3741} Z_i, \quad \text{and} \quad \text{C-index} = \frac{\text{Concordance}}{3741}.$$

Interpretation is straightforward: a C-index of 0.5 or lower indicates predictive performance no better than random chance, while a C-index of 1 indicates perfect ordering of predicted survival times.

2.2 Gradient Boosted Model

Gradient Boosting is another ensemble learning technique that uses sequentially trained decision trees to produce a "strong learner" from a series of weaker machine learning models, performing the partial derivative (hence the name "gradient") of the halved residuals (loss function) at each step to derive an incrementally more accurate training model.

In past literature, Gradient Boosting has been primarily used for binary classification of diseases and identification of species present in biological systems. Its most classical applications come from neural networks, where the process of boosting a model from a series of iterative decision trees is similar to the translation of data through the layers of a multi-modal network.

In this section, we will adapt the binary classifier-based Gradient Boosted machine learning architecture to perform survival time prediction. [13]

2.2.1 Methodology

To run a Gradient Boosted model, specifically a Gradient Boosted model for regression (due to the continuous nature of our data), we must first introduce some notation. Let x_j denote the covariate vector and s_j denote the survival time for the jth patient in our test, for all $j \in \mathbb{Z}^+$ with $j \leq 87$. Then, we define the following loss function most appropriate for manual computation of the prediction estimator $\hat{f}$:

$$\mathcal{L}(s_j, \hat{f}(j)) = \frac{1}{2}(s_j - \hat{f}(j))^2$$

where $\hat{f}$ is a function giving the predicted survival time for patient j. On the first iteration of Gradient Boosted network, we produce a singular leaf node to initialize the sequence, and the value this node takes on is such that the cumulative loss is minimized over the set P of all patients in our training set. This can be computed by setting the partial derivative of the loss function, summed for all $j \in P$, with respect to $\hat{f}(j)$ equal to zero as follows:

$$\begin{aligned}
\hat{f}_0(j) &= \arg\min \sum_{j \in P} \mathcal{L}(s_j, \hat{f}(j)) \Rightarrow \frac{\partial}{\partial \hat{f}(j)} \mathcal{L}(s_j, \hat{f}(j)) \\
&= -(s_1 - \hat{f}(1)) - (s_2 - \hat{f}(2)) - \cdots - (s_{86} - \hat{f}(86)) - (s_{87} - \hat{f}(87)) = 0 \\
&\Rightarrow \hat{f}_0(j) = \frac{1}{87} \sum_{j \in P} s_j
\end{aligned}$$

where, as a reminder, $|P| = 87$ denotes the cardinality of the patient training set for our sample data. Hence, the primary leaf in the Gradient Boosted network comprises the arithmetic mean of the survival times in the patient data, as intuition suggests. Next, we can construct a tree to estimate the residuals of the predicted outputs from the zeroth iteration, programmed from the covariate vectors of the training data via an approximate greedy search algorithm which splits each node from the root node to the penultimate nodes in such a way that maximizes the gain:

$$\text{Gain} = \frac{1}{2}\left(\frac{G_L^2}{H_L + 1} + \frac{G_R^2}{H_R + 1} - \frac{(G_L + G_R)^2}{H_L + H_R + 1}\right).$$

Here, subscripts L and R correspond to the left and right nodes of each candidate split respectively, G and H represent the gradient and hessian of the loss function respectively, and the "+1" term in the denominator serves as a regularization hyperparameter to protect

against overfitting. For instance, considering without loss of generality the root node split $X_1 < k$ for covariate X_1 and $k \in S$, the set of patient survival times, then

$$G_L = \sum_{j \in S_L} \frac{\partial}{\partial \hat{f}(j)} \mathcal{L}(s_j, \hat{f}(j)), \quad H_L = \sum_{j \in S_L} \frac{\partial^2}{\partial \hat{f}(j)^2} \mathcal{L}(s_j, \hat{f}(j))$$

with G_R and H_R computed similarly. Note that performing the gradient calculation to maximize the gain is analogous to computing and summing the negative pseudo-residuals for each training patient at this step, given by:

$$r_0(j) = s(j) - \hat{f}_0(j).$$

Due to the low sample size of our data, we will restrict the number of splits at this tree and all future trees formed to only 8 (marginally greater than a stump) as to mitigate the probability of our model fitting to the nuances in the training data. As a final measure to account for overfitting, we will incorporate a shrinkage rate of $\lambda = 0.05$ into our model, which results in the relation:

$$\hat{f}_n(j) = \hat{f}_{n-1}(j) + \lambda N_{n,j}, \quad 1 \leq n \leq 100$$

where $N_{n,j}$ denotes the terminal node value (or average terminal node value) of patient j, for whom the model predicts survival time after being processed through the nth tree. To establish an equal means of comparison against RSF, we will employ Harrell's C-index again to proxy the predictive accuracy of $\hat{f}_{100}(j)$ on our testing set. However, unlike RSF, we will first randomly split the data with a training set to testing set ratio of $4:1$, and will calculate the C-index on the test set rather than the full set of parent data.

2.3 Results & Interpretation

In Python, we manually implemented a Random Survival Forest model with an out-of-box (OOB) validation set. We also used an extreme form of Gradient Boosting known as XGBoost to derive the results of our regression. Both C-indices are captured below.

Model	**C-index**
RSF	0.6928
XGBoost	0.7143

Table 1. C-indices for ensemble models.

The C-index from our Random Survival Forest model, which is under 0.7, indicates a moderate discriminatory capacity of patient survival time from the six features. XGBoost's C-index, in the range of 0.7-0.8, indicates a good to moderately strong model for predicting patient survival time in the testing set. Clearly, the XGBoost model demonstrated better performance on our dataset, perhaps in-part due to its clean nature and the select few important features included.

To bolster these results, we apply Cramér's ϕ for both models, which measures the association between predicted and real results in the data by assuming it comes from a χ^2 distribution. Since time until death is a quantitative measure and Cramér's ϕ is a measure attached to categorical variables, we first grouped survival time by severity in equal time frames:

- **Terminal**: $<$ 12 months
- **Very High Risk**: 12 - 24 months
- **High Risk**: 24 - 36 months
- **Moderate Risk**: 36 - 48 months
- **Low Risk**: 48 - 60 months
- **Minimal Risk**: 60+ months.

It should be noted that the overall survival data is skewed right even with these categories, resulting in most ensemble learning types, including RSF and XGBoost, to favor predicting terminal results. This byproduct of our low sample size is, however, accounted for by the low number of above groups chosen compared to the number of total data points, which was done to avoid penalizing a model despite making few outlier predictions compared to significantly many accurate ones.

RSF and XGBoost produced ϕ_c values of 0.4188 and 0.5270 respectively, placing RSF in the threshold of moderately strong association and XGBoost in the threshold of strong association. Thus, although both models demonstrate strong capacities for predicting survival time both quantitatively and categorically for brain tumor patients, the evidence from our results suggest that XGBoost is the superior ensemble model.

However, for extremely precise survival time predictions, neither model is an ideal estimator, especially for small datasets such as our sample patient data. Hence, our proxied forecast-accuracy results are intended as a cautionary tale for the careful exercise of machine learning for regression on multiple features, as well as the need for larger open-source datasets to be made available in the realm of brain tumor survival.

Finally, we present the feature importance from both models, comparing them to each other and to the results from our previous paper on parametric models.

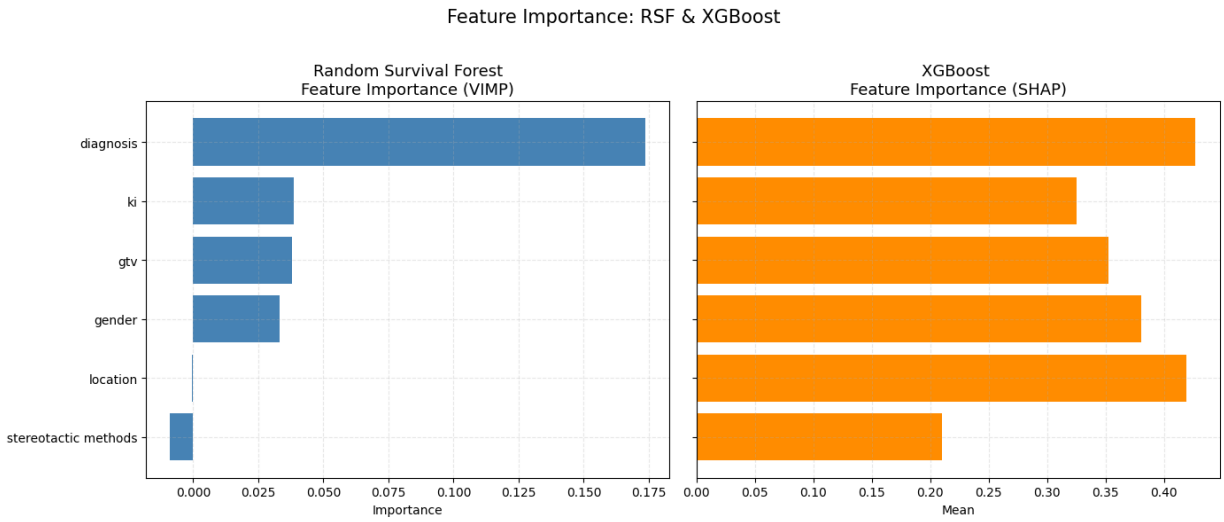

Figure 1. Feature Importance Comparison.

In both models, diagnosis is the most important covariate; however, diagnosis is significantly (at $\alpha = 0.001$) important for RSF as opposed to for XGBoost. With XGBoost, the remaining features are important to roughly similar extents, with stereotactic methods being a slightly less important feature that the rest. In RSF, the inclusion of stereotactic methods is more dramatic; it is the only feature whose inclusion decreases the model's accuracy. The difference between RSF's feature importance and XGBoost's feature importance is plausibly accounted for by the difference in their tree-depth; XGBoost's comparatively low tree depth minimizes its model variance, making it less susceptible to high skew from extremely significant features like diagnosis.

In the second part of this study, we employed a log-rank test to determine the significance of groups within each feature, a measure analogous to feature importance. The study concurred with the critical importance of diagnosis, which had a log-rank p-value several orders of magnitude higher than any other covariate, and agreed with RSF regarding the

lack of evidence supporting the importance of gender or location in survival time prediction. However, the previous study found stereotactic methods to be significant at $\alpha = 0.05$ (but not $\alpha = 0.01$), which starkly differs from our ensemble results here. [2] Hence, just as some medical journals claim SRT is superior to SRS while others claim that the difference is negligible, our models disagree in a way which highlights the discrepancy. Lastly, to reiterate, the previous study used parametric models to predict survival function, while this study uses non-parametric models to predict survival time; differences in feature importance can be accounted for by the difference in these functions.

3 Summary and Conclusion

Although the difference between ensemble models is not stark for the low sample size of our data, the results concur with the superior predictive capacity of XGBoost in the hierarchy of ensemble learning models from several previous sources of literature.

Within reputed literature, many of the studies that relied solely on clinical, MRI, or demographic data produced models with mediocre accuracies. Specifically, accuracy with RSF was frequently achieved in the 50-60th percentile, while accuracy with Gradient Boosting was up to 10% higher but still not remarkable. Compared to other models trained on open datasets, many of which only contained MR images, our model achieved a C-index typically 0.01-0.1 points higher in magnitude. This superior performance demonstrates the importance of clinical data consideration in improving model accuracy.

The literature differed, as with our previous paper, in the ranking of features used to train each model. However, most RSF and Gradient Boosting models determined that Karnofsky index was among the most important features, as well as mutation type - which is similar to diagnosis - or diagnosis directly. Thus, our models align with literature in regard to the importance of diagnosis, with RSF overplaying its importance. Meanwhile, our RSF more closely captures the relative importance of the Karnofsky index measure copmared to the literature than XGBoost.

In the fields of radiology and oncology, particularly in the context of providing palliative care for brain tumor patients, survival analyses is integral. Providing patients who wish to know their life expectancy with a clear and accurate survival time estimate allows them and their loved ones to optimize the time that they have to spend together, as well as mitigates the extent of shock-induced grief associated with their passing. Such estimates have typically been avoided due to human inaccuracies; employing ensemble learning models

such as ours, especially when trained on more plentiful and diverse datasets, can eliminate accuracy concerns and allow for a more confident timeline to be catered to patients.

4 Future Works

It could be worth implementing other models such as Support Vector Machine (SVM) or K-Nearest Neighbors (KNN). Although these models are traditionally out-performed by RSF and XGBoost, in situations where computational power is not a concern, combining supervised models and aggregating their results may produce an increased prognosis accuracy.

Additionally, incoporating omic data alongside clinical data is likely to increase accuracy. Oncologists with access to cancer labs should more often, with patient consent, perform tests characterizing the genomes and proteomes of the brain tumors of those inducted into studies like that of Masaryk. This data is much more specific that a simple diagnosis type, and coupled with the patient data we investigated in our study, can significantly boost survival models' accuracy. More clinical data, such as patient age, can easily be collected and should be incorporated into open-source datasets as well.

Lastly, our models should be used in collaboration with classification models. In our first paper from this three-part series, we investigated employing Convolutional Neural Networks (CNNs) to classify brain tumors. [14] Models trained using CNNs can diagnose patients by extracting patterns from their MRI scans, and this diagnosis can then be fed into an XGBoost model alongside other data points to predict patient survival. Combining classification and survival models into a single step can streamline the prognosis process, resulting in greater assistance to oncologists in determining the severity of a patient's condition.

Supplemental Materials

Supplemental materials, including the dataset and Python code used for analysis, are available at the following GitHub repository.

https://github.com/Aryan-Mukherjee007/BrainTumorEnsemble.

Readers are encouraged to reproduce the results and explore the data further.

Acknowledgements

I would like to sincerely thank Dr. Olga Korosteleva, Professor of Statistics at California State University, Long Beach, for her invaluable guidance, support, and encouragement throughout this research. I am also grateful to Mr. Eric Shulman, mathematics teacher at University High School, for his instruction in calculus and matrix theory, which laid foundational knowledge essential to this work. Finally, I deeply appreciate my family for their support and patience, which made this journey possible.

References

[1] Ostrom, Q. T., Price, M., Neff, C., Cioffi, G., Waite, K. A., Kruchko, C., and J. S. Barnholtz-Sloan. (2024). CBTRUS Statistical Report: Primary Brain and Other Central Nervous System Tumors Diagnosed in the United States in 2017-2021. *Neuro-Oncology*, 26 (Supplement 6): vi1-vi85. doi: 10.1093/neuonc/noae145.

[2] Mukherjee, A. (2025). Prognostic Modeling of Brain Tumor Survival. *Intelligence Planet*, 2(3).

[3] Selingerová, I., Doleželová, H., Horová, I., Katina, S., and J. Zelinka. (2016). Survival of patients with primary brain tumors: comparison of two statistical approaches. *PLOS ONE*, 11(2): e0148733. doi:10.1371/journal.pone.0148733. PMID: 26863415; PMCID: PMC4749663.

[4] Kim, S., Luna, M., Chikontwe, P., and S. Park. (2020). Two-Step U-Nets for Brain Tumor Segmentation and Random Forest with Radiomics for Survival Time Prediction. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Lecture Notes in Computer Science, vol. 12263: 156–165. doi:10.1007/978-3-030-46640-4_19.

[5] Qiu, X., Gao, J., Yang, J., Hu, J., Hu, W., Kong, L., and J. J. Lu. (2020). A Comparison Study of Machine Learning (Random Survival Forest) and Classic Statistic (Cox Proportional Hazards) for Predicting Progression in High-Grade Glioma after Proton and Carbon Ion Radiotherapy. *Frontiers in Oncology*, 10: 551420. doi:10.3389/fonc.2020.551420.

[6] Audureau, E., Chivet, A., Ursu, R., Corns, R., Metellus, P., Noel, G., Zouaoui, S., Guyotat, J., Le Reste, P. J., Faillot, T., Litre, F., Desse, N., Petit, A., Emery, E., Lechapt-Zalcman, E., Peltier, J., Duntze, J., Dezamis, E., Voirin, J., Menei, P., Caire, F., Dam Hieu, P., Barat, J. L., Langlois, O., Vignes, J. R., Fabbro-Peray, P., Riondel, A., Sorbets, E., Zanello, M., Roux, A., Carpentier, A., Bauchet, L., and J. Pallud. (2018). Prognostic factors for survival in adult patients with recurrent glioblastoma: a decision-tree-based

model. *Journal of Neuro-Oncology*, 136(3): 565–576. doi:10.1007/s11060-017-2685-4.

[7] Karami, G., Giuseppe Orlando, M., Delli Pizzi, A., Caulo, M., and C. Del Gratta. (2021). Predicting Overall Survival Time in Glioblastoma Patients Using Gradient Boosting Machines Algorithm and Recursive Feature Elimination Technique. *Cancers (Basel)*, 13(19): 4976. doi:10.3390/cancers13194976.

[8] Rajpurkar, S., Agravat, R., Roy, M., and M. S. Raval. (2021). Glioblastoma Multiforme Patient Survival Prediction. *arXiv preprint* arXiv:2101.10589.

[9] Charlton, C. E., Poon, M. T. C., Brennan, P. M., and J. D. Fleuriot. (2023). Development of prediction models for one-year brain tumour survival using machine learning: a comparison of accuracy and interpretability. *Computer Methods and Programs in Biomedicine*, 233: 107482. doi:10.1016/j.cmpb.2023.107482.

[10] Renugadevi, M., Anitha, J., Dhanasekaran, S., Karthik, S., and M. Devasia. (2023). Machine Learning Empowered Brain Tumor Segmentation and Grading Model for Lifetime Prediction. *IEEE Access*, 11: 120868–120880. doi:10.1109/ACCESS.2023.3326841.

[11] El Hachimy, I., Benamar, N., and N. Hajji. (2024). Machine Learning and Omics Data for Predicting Overall Survival in Glioblastoma Patients. *2024 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, Sakhir, Bahrain, pp. 66–71. doi:10.1109/3ict64318.2024.10824342.

[12] Hartshorn, S. (2016). *Machine Learning with Random Forests and Decision Trees: A Mostly Intuitive Guide, But Also Some Python*. Independently published.

[13] Wade, C., and K. Glynn. (2020). *Hands-On Gradient Boosting with XGBoost and scikit-learn: Perform accessible machine learning and extreme Gradient Boosting with Python*. Packt Publishing.

[14] Mukherjee, A. (2025). Brain Tumor Diagnosis Using Convolutional Neural Network on Magnetic Resonance Imaging Data. *Intelligence Planet*, 2(2).

Appendix

Data Frequency Charts

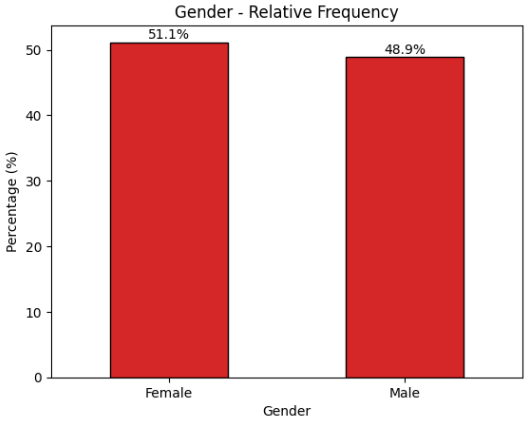

Figure 1: Gender

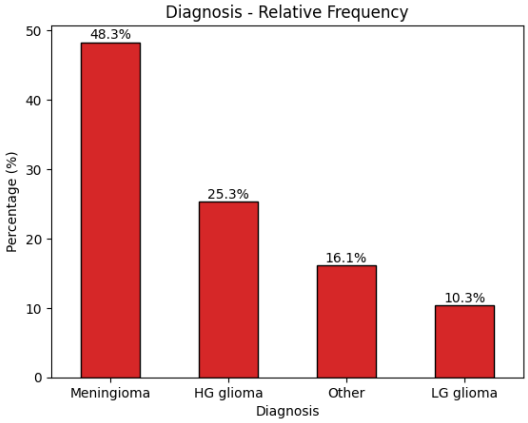

Figure 2: Diagnosis

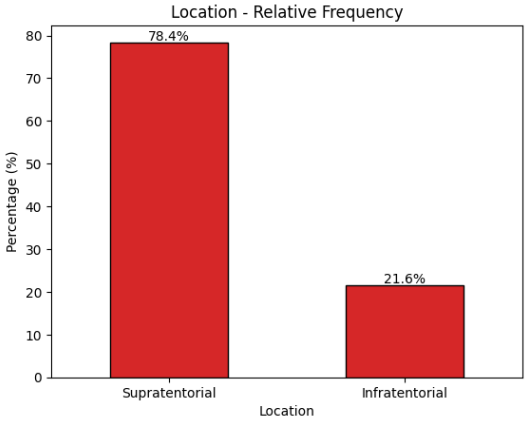

Figure 3: Location

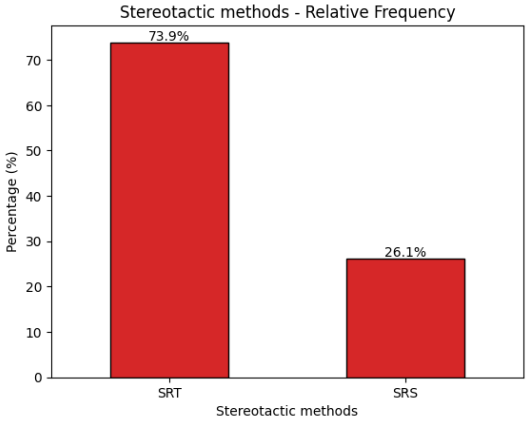

Figure 4: Stereotactic Methods

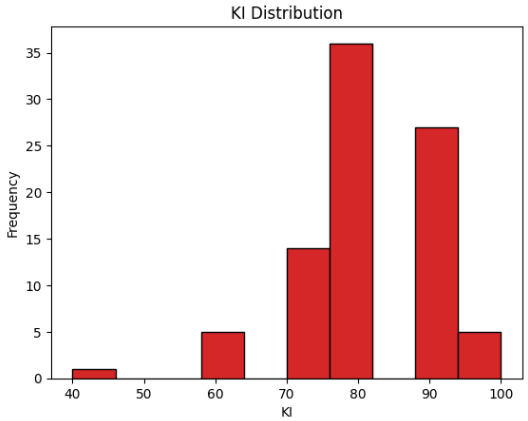

Figure 5: Karnofsky Index

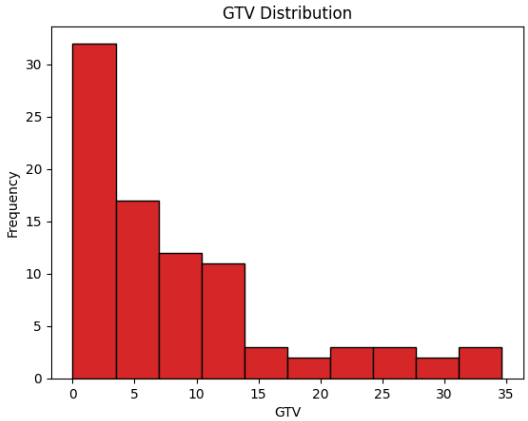

Figure 6: Gross Tumor Volume