Glioma Grading

April, 2025

Abstract

Gliomas are the most common primary tumors of the brain with two main types: LGG (Lower-Grade Glioma) or GBM (Glioblastoma Multiforme). The two types of tumors have different characteristics and therefore require different treatment and care. Differentiating between the two types allows healthcare providers to better optimize treatment, decrease treatment toxicity, and increase survival rates. In the past, classifying the tumors (also known as grading) involved simply looking at the cells under a microscope; however, in recent years this process has developed to include testing genetic features. Our goal will be to create a classification model to predict the grade of tumor and identify the most impactful genetic and clinical features for prediction.

Stat 432 Final Report

Name: Karina Grewal **NetID:** kgrewal2

Contributions: Worked on data pre-processing, visualizations and 2/6 model implementations. Worked equally on writing the report.

Name: Steve Liang **NetID:** sliang3

Contributions: Worked on 4/6 model implementations. Worked equally on writing the report.

Contents

1 Introduction

Our objective is to determine whether a patient is LGG or GBM with given clinical and genetic/molecular features. Low-grade gliomas are cancerous brain tumors that arise from the support cells (glial cells) within the brain. They are similar to glioblastomas, but are slow growing, and only make up 20 percent of all primary brain tumors. [3] Glioblastomas are common primary brain tumors that start within the brain and quickly spread throughout the brain. This type of cancerous tumor is much more deadly and aggressive to the LGG type. It is important to identify these cases and provide care in a timely manner. Glioblastomas arise from the cells that support the neurons within the brain. While it can be located anywhere in the brain, it is commonly found in the cerebral hemisphere near the frontal and temporal lobes. [4]

The main task is to find the best classification model and identify the most important features. We will apply different classification methods such as KNN, Logistic Regression, Random Forest, Gradient Boosting, AdaBoost, and SVM and then choose the model that performs best. Through our analysis, we will also be able to understand which features are the most important in differentiating between the tumor types by extracting the feature importances from our models. Our end goal is to help healthcare professionals more accurately diagnose GBM cases and to provide a better understanding of the genetic features of gliomas for further research, helping healthcare professionals provide better care to patients.

1.1 Data Source

Our dataset is the Glioma Grading Clinical and Mutation Features dataset from the UCI machine learning repository, with the data originally coming from The Cancer Genome Atlas (TCGA) program. The most frequently mutated 20 genes and 3 clinical features are considered from TCGA-LGG and TCGA-GBM brain glioma projects.[1]

1.2 Plan of Analysis

We will clean the data, deal with any missing values or anomalies, and perform any necessary feature engineering in order to feed the data into our chosen models. We will create an 80/20 train-test split for our data for all the models with all model evaluation performed on the same test subset of the dataset.

To measure and compare the performance of our models, we will choose a metric that prioritizes sensitivity for predicting GBM. Because GBM is more aggressive, we believe it is more important to accurately identify cases of GBM. For our second objective in aiding further research for understanding the relationship between genetic features and the type of tumor, we will choose a metric that prioritizes overall model performance such as AUC or F1 score and identify the most important factors in prediction.

2 Data Cleaning

In this section, we will explore the raw dataset. From there, we will discuss data cleaning techniques performed on the raw data, and highlight our initial data observation findings. The exact code can be found in Section 8.1.

2.1 Data Overview

The dataset contains 27 variables in total. Of the 27 variables, there are 20 genes, 3 clinical features, 3 identification, and 1 output columns. We will be dropping the identification columns as they do not bring any purpose to the goal at hand. Of the 23 explanatory variables, there is 1 quantitative variable (Age), and 21 factor variables (including race, gender, and mutated genes). The output variable is categorical, classifying the tumor as LGG or GBM. There are a total of 862 observations in the dataset. There are 23 missing data values and as such we used imputation to preserve the information in those points.

Table 1: Clinical, Molecular, and Class Variables in Glioma Dataset with Value Encodings

#	Type	Name	Value Encoding
1	Clinical	Gender	0 = Female 1 = Male
2	Clinical	Age at Diagnosis	Continuous (years)
3	Clinical	Race	0 = White 1 = Black or African American 2 = Asian 3 = American Indian or Alaska Native
4–23	Molecular	Gene Mutations (IDH1, TP53, ..., PDGFRA)	0 = Not-Mutated 1 = Mutated
24	Class	Tumor Grade	LGG = Low-grade glioma GBM = Glioblastoma (high-grade)

2.2 Data Cleaning

As a part of the pre-processing stage, the Age At Diagnosis feature values were converted from string to continuous value. The Age of each patient was originally recorded as number of years and days. We aggregated the information by converting the number of days into years and adding it to the corresponding number of years for that patient.

Also, the dataset includes rows with missing data. There are 23 instances in the original file where Gender, Age at Diagnosis, or Race feature values are '–', or 'not reported' We will impute the missing data values, by extrapolating information from it's column.

2.2.1 Imputing Columns with NA Values

Our approach to handling missing data values is to take the median and mode of each of the columns. First, we replaced the '–', or 'not reported' with NA and then continued with imputation. For the two categorical variables, we used their mode to replace the NA values with the most common value. For the variable Age at Diagnosis, we used the median value to replace the NA's. After this step, we no longer have any missing data.

3 Data Analysis

We visualized the cleaned data to obtain an understanding of the distribution of variables. The code can be found in Section 8.2. To do so, we plotted the Grade of tumour across variables such as Age, Gender and whether a specific gene had mutated. In doing so, we hope to obtain a rudimentary understanding of the relationships between each individual explanatory variable and Grade.

3.1 Quantitative Variables

Figure **1** plots the quantitative variable Age at Diagnosis against the categorical variable Grade. From the boxplot we can observe the following:

- The distribution of Age seems to have little overlap between different Grades. It is hard to determine at this stage if there is a statistically significant difference.
- We see that there is a lot of overlap between Genders for each Grade category in comparison to Age.

Figure 1: Grade and Gender Variables Plotted Against Age at Diagnosis

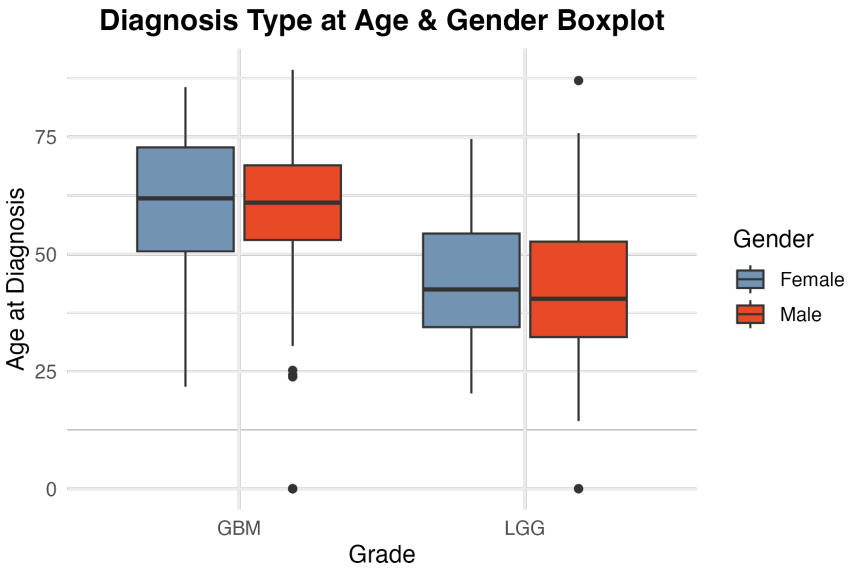
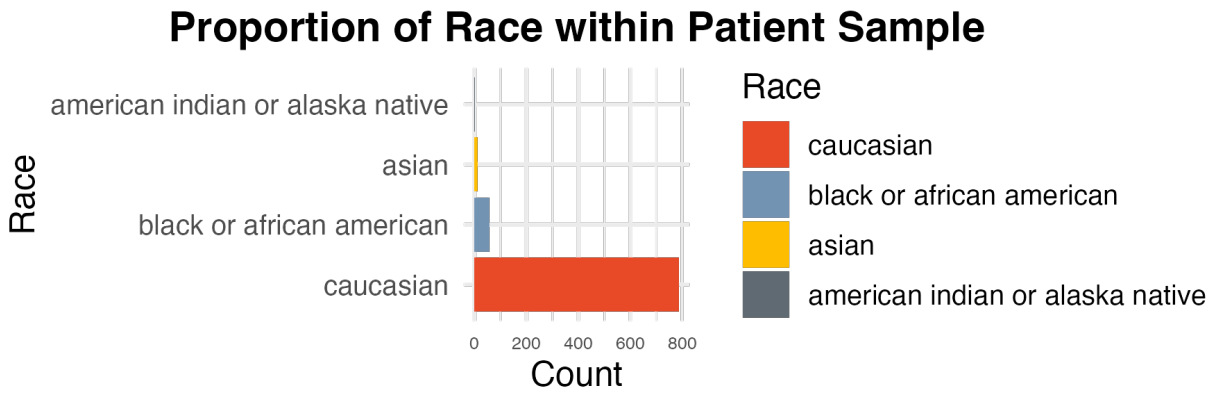


Figure 2: Proportion of Race within Patients



3.1.1 Proportion of Race within Patients

We also created a plot of the demographics of our sample population in figure **2**. We observed that there is an overwhelming number of Caucasian patients in this sample so our results may be biased towards that race.

3.1.2 Proportion of Classifications within Genes

Here, we have plotted the counts of 20 genes within our sample of patients in figure **3**. A 0 indicates that gene has not mutated and a 1 indicates the gene has mutated inside the patient. Within each count there is a colour coded proportion of how many patients were diagnosed with which Grade of tumour. We observe the following:

- It's clear that for most patients the majority of the genes (excluding IDH1 and TP53) did not mutate.

Figure 3: Proportion of Classifications within Genes

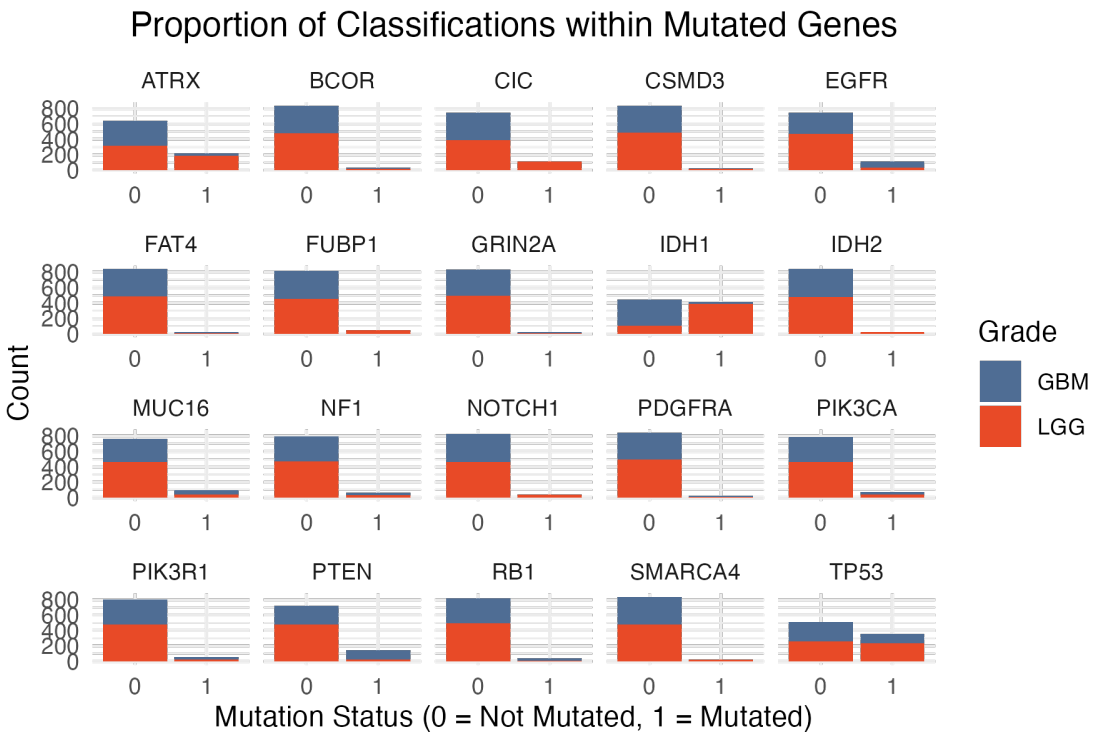
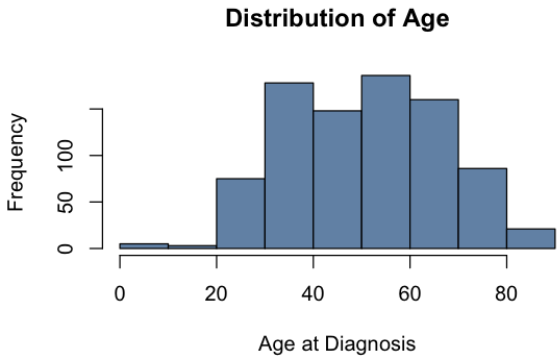


Figure 4: Distribution of Age at Diagnosis



- Within the group of patients who's gene did not mutate (excluding IDH1 and TP53), it looks as if half were diagnosed with a LGG tumour and the other half with GBM.

3.1.3 Age of Diagnosis Distribution

In figure **4**, we have plotted the histogram for the distribution of Age at Diagnosis. We see there is a rough bell shape to the distribution and therefore we do not need to consider any transformation to this variable. We see that most patients are between 20 and 80 years old.

4 Model Implementation

In this section, we will discuss the six models we will use and how they are applicable to our dataset. We will also discuss any parameter tuning used. Tuning and model implementation code can be found in Section 8.3

4.1 KNN Classifier

The KNN classifier predicts glioma grade by identifying the k most similar training cases based on Euclidean distance in the feature space. We used cross-validation methods to find the optimal value of k by minimizing test error. While this model is simple and does not require any training or assumption on the data distribution, it is sensitive to irrelevant features (due to curse of dimensionality). It also does not handle imbalanced classes well. KNN is a good benchmark model to compare to other models (Random Forest and SVM) that can handle higher dimensional data better.

Tuning Process: Tuning k in KNN involves selecting the optimal number of neighbors to balance bias and variance. A small k may overfit noise, while a large k can oversmooth decision boundaries. We used 20-fold cross-validation to find the k that maximizes AUC on validation data. We tuned k using the following values: 1, 5, 10, 20, 50, 100. The optimal k was found to be 10.

4.2 Logistic Regression

Logistic Regression is a simple model used for binary classification. It predicts the probability of an outcome using the logistic function, which maps input features to a value between 0 and 1. Unlike linear regression, it uses maximum likelihood estimation to fit the model. Key outputs include coefficients (log-odds) for each feature, showing their impact on the outcome. It's simple, interpretable, and works well for linearly separable data.

4.3 Random Forest

Random Forest is an ensemble method that uses multiple decision trees trained on bootstrapped samples with random feature selection at each node. This model handles high dimensional data well and can capture non-linear relationships between variables.

Tuning Process: We tuned the mtry parameter, which is the number of nodes randomly considered for splitting in each tree. The optimal value for this parameter was 4.

4.4 Gradient Boosting

Gradient Boosting is a powerful machine learning ensemble method that combines multiple weak learners to create a strong predictive model. It works by sequentially adding models that focus on correcting errors made by previous models, ultimately improving accuracy and reducing bias. Boosting has very high predictive power, but can be prone to overfitting if parameters are not tuned carefully.

Tuning Process: We tried 3 different learning rates (0.1, 0.01, 0.001) and 3 tree depths (1,2,3) and used cross-validation to find the optimal number of trees on each combination. We landed on a model with a learning rate of 0.01 and tree depth 3.

4.5 AdaBoost

Similar to Gradient Boosting, AdaBoost is a variation of boosting that adaptively adjusts the weight iteratively during training.

Tuning Process: We landed on the same learning rate and tree depth as the gradient boosting machine and used cross-validation to find the optimal number of trees

4.6 SVM

A Support Vector Machine is an algorithm that works by finding the optimal hyperplane that separates data points into different classes. It is a method that works well for high-dimensional data and can handle both linear and non-linear relationships.

Tuning Process: We tested models with radial and polynomial kernels, as well as a linear model for a baseline. We chose the best model using a range of cost parameters (0.001, 0.01, 0.1, 1, 5, 10, 100) and additionally degrees (2, 3) for the polynomial model. The optimal radial model had cost = 10 and the polynomial had cost = 100 and degree = 2. Both models performed very similarly in accuracy and AUC. We selected the radial model due to its slightly better performance.

5 Model Evaluation

To measure the classification power of our 6 models, we used 6 evaluation metrics: classification accuracy rate, Area Under the ROC Curve (AUC), F-Measure, precision, recall, and specificity. We will discuss each of these metrics in this section.

5.1 AUC

AUC, the area under the ROC curve is calculated by first plotting the true positive rate against the false-positive rate for the performance of the classification model. A perfect model that classifies everything correctly will have an AUC value of 1. While the worst model that did not classify anything correctly will have an AUC value of 0. An AUC value of 0.5 indicates that the model is no better than simply classifying the points randomly.

5.2 Performance Metric Definitions

We use the following abbreviations and their applications to our case:

- **TP (True Positive):** Positive class is classified as positive.
  - High grade glioma (GBM) is classified as a high grade (GBM)
- **TN (True Negative):** Negative class is classified as negative.
  - Low grade glioma (LGG) is classified as a low grade (LGG)
- **FP (False Positive):** Negative class is misclassified as positive.
  - Low grade glioma (LGG) is classified as a high grade (GBM)
- **FN (False Negative):** Positive class is misclassified as negative.
  - High grade glioma (GBM) is classified as a low grade glioma (LGG)

Table 2: Performance Metrics Overview with Clinical Implications

Metric	Definition and Clinical Consequence	Formula
Accuracy	Proportion of all correct predictions (TP + TN). **Consequence**: Provides an overall diagnostic reliability.	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision	Ratio of true positives to all positive predictions. **Consequence**: High precision reduces unnecessary toxic/invasive treatments for false GBM diagnoses (FP), but may miss true GBMs (FN).	$\frac{TP}{TP+FP}$
Specificity	Proportion of LGG (negative) cases correctly identified. **Consequence**: High specificity prevents over-treatment of LGG as GBM (FP), but poor specificity risks toxic treatment for low grade cases when not necessary.	$\frac{TN}{TN+FP}$
Sensitivity	Proportion of GBM (positive) cases correctly detected. **Consequence**: High sensitivity minimizes deadly missed GBM diagnoses (FN), but may increase false alarms (FP). Higher sensitivity is important for classifying aggressive tumors.	$\frac{TP}{TP+FN}$

6 Results

In this section we will discuss the results from the aforementioned methods.

6.1 Compiled Model Metrics

Table 3: Model Performance Comparison

Model	AUC	Accuracy	Sensitivity	Specificity	Precision	F1 Score
KNN	0.8793	0.8547	0.7887	0.9010	0.8485	0.8175
Logistic Regression	0.9092	0.8547	0.7662	**0.9263**	**0.8939**	0.8252
Random Forest	0.8963	0.8488	**0.9091**	0.8113	0.7500	0.8219
Gradient Boosting	0.9119	**0.8721**	0.8585	0.8939	0.7973	**0.8268**
AdaBoost	0.9141	0.8605	0.8396	0.8939	0.7763	0.8067
SVM	**0.9202**	0.8488	0.8019	0.9242	0.7439	0.7718

Note: Bold values are the best results.

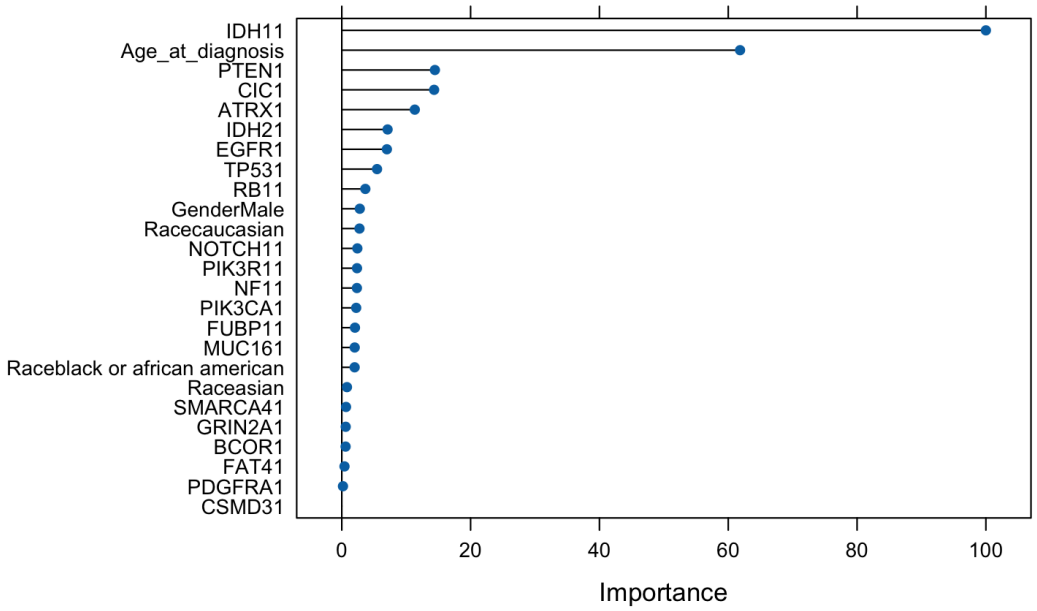
The comparative analysis of six machine learning models revealed that SVM achieved the highest AUC (0.9202), followed closely by AdaBoost (0.9141) and Gradient Boosting (0.9119). Gradient Boosting demonstrated the best accuracy (0.8721), while Random Forest showed the best sensitivity (0.9091) at the cost of lower specificity (0.8113). Notably, Logistic Regression achieved the highest precision (0.8939) and the best specificity (0.9263), suggesting particularly reliable positive predictions when it identified the negative class effectively. KNN delivered balanced performance across metrics, though it ranked lowest in AUC among all models.

The F1-score analysis, which balances precision and sensitivity, found Gradient Boosting as the top performer (0.8268), with Logistic Regression (0.8252) and Random Forest (0.8219) following behind. This indicates these models best handled the trade-off between identifying true positives and minimizing false

alarms (false positives). Surprisingly, despite SVM's strong AUC, it yielded the lowest F1-score (0.7718), highlighting a precision-sensitivity imbalance. The results suggest Gradient Boosting as the most robust overall model, while Logistic Regression may be preferable when high precision is prioritized. All models achieved comparable accuracy (0.8488–0.8721), confirming consistent predictive capability across different algorithmic approaches.

6.2 Variable Importance

Figure 5: Variable Importance



Here is the variable importance graph as per the Random Forest Model. Across all models we found that the gene IDH1 and Age At Diagnosis were the most important (or significant) variables when predicting whether a tumour should be classified as GBM or LGG. Both the Gradient Boosting and AdaBoost models output a similar result for variable importance.

7 Discussion & Conclusions

For the purpose of assisting healthcare professionals in classifying between GBM and LGG, we would prioritize the sensitivity metric and therefore recommend the Random Forest model. In the case of supporting further research in understanding the relationship between genetic features and tumor types, we would prioritize overall performance and recommend the Gradient Boosting model which performed best in accuracy and F1 score. For our models that could output feature importance, the two most important features were the molecular marker IDH1 and Age. This is completely expected, as IDH is the main molecular marker that is tested to grade gliomas.[6] A potential application for our work in predicting the grade of gliomas based on genetic/molecular features can be to help researchers identify and prioritize which molecular markers to test [2]. While this is already being done for glioma grading, our models can potentially contribute to similar endeavors for diagnosing other diseases.

7.1 Limitations

- The dataset overrepresented Caucasian patients as seen in figure **2**, which may limit generalizability to more diverse populations. Racial disparities in genetic markers or treatment responses could skew model performance in real-world settings.
- Clinical features (e.g., tumor location, treatment history) were limited. Including such data might improve model robustness.
- While SVM and Gradient Boosting performed well, their interpretation is hard to mesh into a clinical setting. Logistic Regression's interpretability (via coefficients) could make it preferable in practice, despite slightly lower AUC.

7.2 Future Investigation

- Any future investigation would ideally involve consultation with experts in this field to provide additional background knowledge and to help identify areas where we can make the most impact. For example, they could guide us towards ongoing research towards understudied markers that can help enhance predictive power.
- To mitigate bias, we can collect more data or merge data with medical centers from around the world to create a more racially diverse dataset. In that way, our findings can be more universally applicable. Techniques such as adversarial debiasing could also be explored.
- Introducing a longitudinal study that can help track the models performance over time to assess how the level of tumour progression (LGG progressing into a GBM) can affect prediction.

8 Appendices

8.1 Data Preprocessing

```
#define a function to replace '--' and 'not reported' with NA
replace_missing <- function(x) {
  x[x == "—" | tolower(x) == "not reported"] <- NA
  return(x)
}

#apply the function to relevant columns
glioma_labels$Gender <- replace_missing(glioma_labels$Gender)
glioma_labels$Race <- replace_missing(glioma_labels$Race)
glioma_labels$Age_at_diagnosis <- replace_missing(glioma_labels$Age_at_diagnosis)

glioma_labels <- glioma_labels %>%
  mutate(
    #extract number of years and days (age) of patient
    years = as.numeric(str_extract(Age_at_diagnosis, "\\d+(?=\\s*years)")),
    days = as.numeric(str_extract(Age_at_diagnosis, "\\d+(?=\\s*days)")),

    #replace NA with 0 if one of the parts is missing
    years = ifelse(is.na(years), 0, years),
    days = ifelse(is.na(days), 0, days),

    #convert to decimal years
    Age_at_diagnosis = years + (days / 365)
  ) %>%
  select(-years, -days)

#impute Gender with mode
glioma_labels$Gender[is.na(glioma_labels$Gender)] <-
    names(sort(table(glioma_labels$Gender), decreasing = TRUE))[1]

#impute Race with mode
glioma_labels$Race[is.na(glioma_labels$Race)] <-
    names(sort(table(glioma_labels$Race), decreasing = TRUE))[1]

#convert Age_at_diagnosis to numeric
glioma_labels$Age_at_diagnosis <-
    as.numeric(glioma_labels$Age_at_diagnosis)

#impute Age_at_diagnosis with median
glioma_labels$Age_at_diagnosis[is.na(glioma_labels$Age_at_diagnosis)] <-
    median(glioma_labels$Age_at_diagnosis, na.rm = TRUE)

# Check for any remaining missing values
sum(is.na(glioma_labels$Gender))
sum(is.na(glioma_labels$Race))
sum(is.na(glioma_labels$Age_at_diagnosis))

#hot coding mutation columns
```

```
mutation_cols <- colnames(glioma_labels)[8:27]

glioma_labels <- glioma_labels %>%
  mutate(across(all_of(mutation_cols),
                ~ ifelse(. == "MUTATED", 1,
                         ifelse(. == "NOT_MUTATED", 0, NA))),
         across(8:27, as.factor))

#turning hotcoded columns into factor cols
glioma_labels <- glioma_labels %>%
  select("Age_at_diagnosis", everything()) %>%
  mutate(across(2:24, as.factor))
```

8.2 Plots

Creating age/gender boxplot

```
    p <- ggplot(glioma_labels, mapping = aes(y = Age_at_diagnosis, x = Grade,
                                             fill = Gender)) +
  geom_boxplot() +
  scale_fill_manual(values = c("Male" = "#E84A27", "Female" = "#7393B3")) +
  labs(y = "Age at Diagnosis", title = "Diagnosis Type at Age & Gender Boxplot",
       fill = "Gender") +
  theme(plot.title = element_text(hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5))
```

Creating age histogram plot

```
hist(glioma_labels$Age_at_diagnosis,
     main = "Distribution of Age",
     xlab = "Age at Diagnosis",
     col  = "#7393B3")
```

Creating race barplot

```
race_df$Race <- factor(glioma_labels$Race,
                       levels = rev(levels(factor(glioma_labels$Race))))

race_plot <-  ggplot(race_df, aes(x = Race, fill = Race)) +
  geom_bar(position = 'dodge') +
  labs(fill = "Race", y = "Count", title = "Proportion of Race within Patient Sample") +
  theme(axis.text.x = element_text(size = 6)) +
  theme(plot.title = element_text(hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5)) +
  coord_flip() +
  scale_fill_manual(values = uiuc_extended)
```

Creating barplots for mutates/non-mutated genes

```
uiuc_org <- "#E84A27"
uiuc_blue <- "#4F6D94"
# Convert to long format
```

```
df_long <- glioma_labels %>%
  pivot_longer(cols = c(colnames(glioma_labels)[8:27]),
               names_to = "Feature",
               values_to = "Value")

#grade across each Feature
grade_gene <- ggplot(df_long, aes(x = Value, fill = Grade)) +
  geom_bar(position = "stack") +
  facet_wrap(~ Feature, scales = "free_x") +
  labs(x = "Mutation Status (0 = Not Mutated, 1 = Mutated)",
       y = "Count",
       fill = "Grade",
       title = "Proportion of Classifications within Mutated Genes") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
        plot.subtitle = element_text(hjust = 0.5)) +
  scale_fill_manual(values = c('LGG' = uiuc_org, 'GBM' = uiuc_blue))
```

8.3 Model Tuning

KNN

```
# parameters for cross-validation
resamplingMethod <- "cv"
numFolds <- 20
classificationMethod <- "knn"
performanceMetric <- "ROC"

# sequence of K values
K_seq <- data.frame(k = 1:50)

# setup the cross-validation options
knn_cv_train_control <- trainControl(method = resamplingMethod,
                                     number = numFolds,
                                     classProbs = TRUE,
                                     summaryFunction = twoClassSummary)

# train the model
knn_cv_train <- train(x = X_train,
                      y = Y_train,
                      method = classificationMethod,
                      metric = performanceMetric,
                      trControl = knn_cv_train_control,
                      tuneGrid = K_seq)

print(knn_cv_train)
```

Random Forest Classifier

```
rf_control <- trainControl(method="cv",
number = 10,
search = "grid",
```

```
classProbs = TRUE,
summaryFunction = twoClassSummary)

tunegrid <- expand.grid(.mtry=c(1:6))
Glioma_rfc <- train(Grade~.,
data=Glioma_train,
method="rf",
metric="ROC",
tuneGrid=tunegrid,
trControl=rf_control)
```

Gradient Boosting Model

```
# Learning Rate 0.1
Glioma_GBM1 <- gbm(Grade ~.,
data=Glioma_train_boosting,
distribution="bernoulli",
n.trees=1000,
interaction.depth = 3,
shrinkage=0.1,
cv.folds = 5)
#finding optimal number of trees
best_iter1 <- gbm.perf(Glioma_GBM1, method = "cv")

# Learning Rate 0.01
Glioma_GBM2 <- gbm(Grade ~.,
data=Glioma_train_boosting,
distribution="bernoulli",
n.trees=3000,
interaction.depth = 3,
shrinkage=0.01,
cv.folds = 5)
best_iter2 <- gbm.perf(Glioma_GBM2, method = "cv")
```

AdaBoost

```
Glioma_Ada <- gbm(Grade ~.,
data=Glioma_train_boosting,
distribution="adaboost",
n.trees=3000,
interaction.depth = 3,
shrinkage=0.01,
cv.folds = 5)
best_iter_ada <- gbm.perf(Glioma_Ada, method = "cv")
```

SVM

```
tune_radial=tune(svm,
Grade~.,
data=Glioma_train,
kernel ="radial",
ranges =list(cost=c(0.001 , 0.01, 0.1, 1,5,10,100)),
probability = TRUE
```

```
)
Glioma_SVM_radial <- tune_radial$best.model

tune_poly=tune(svm,
Grade~.,
data=Glioma_train,
kernel ="polynomial",
ranges=list(cost=c(0.001 , 0.01, 0.1, 1,5,10,100),
degree=c(2,3)),
probability = TRUE
)
Glioma_SVM_poly <- tune_poly$best.model
```

9 Bibliography

References

[1] https://archive.ics.uci.edu/dataset/759/glioma+grading+clinical+and+mutation+features+dataset

[2] Hierarchical Voting-Based Feature Selection and Ensemble Learning Model Scheme for Glioma Grading with Clinical and Molecular Characteristics

https://www.mdpi.com/1422-0067/23/22/14155

[3] https://www.mountsinai.org/care/neurosurgery/services/brain-tumors/what-are/low-grade-gliomas

[4] https://www.mountsinai.org/care/cancer/services/brain/tumor/glioblastoma

[5] https://www.hopkinsmedicine.org/health/conditions-and-diseases/gliomas

[6] https://www.mdanderson.org/cancerwise/glioma-vs--glioblastoma--what-is-the-difference-in-these-brai
h00-159537378.html#:~:text=Slower%20growing%20gliomas%20are%20also,aggressive%20the%
20brain%20cancer%20is.