Brain Tumor Prediction

Name:	**Avinash Kumar Kashyap**
Registration No./Roll No.:	21064
Institute/University Name:	IISER Bhopal
Program/Stream:	DSE
Problem Release date:	August 17, 2023
Date of Submission:	November 19,2023

1 Problem Statement

Gliomas are the most common primary brain tumors and can be graded as Lower-Grade Glioma (LGG) or Glioblastoma Multiforme (GBM) based on histological, imaging, clinical, and molecular criteria. Accurate grading is crucial for treatment decisions, but molecular tests are expensive. The goal is to predict glioma grades using a subset of molecular mutation and clinical features, improving accuracy while reducing costs.

2 Introduction

In our project, we are dealing with a dataset containing information about brain tumors. This dataset has 24 different characteristics for 775 patients. Among them, 449 patients have a specific type called Lower-Grade Glioma (LGG), and 326 have Glioblastoma Multiforme (GBM). Initially, we faced a challenge of missing values, but we successfully handled them by either filling in the missing data or using a technique called one-hot encoding.

3 Methods

We have divided our task into three parts

3.1 Data Pre-processing

We first handle missing values by replacing '-' with None and filling missing values with the mode for 'Gender' and 'Race' columns. For 'Age', we convert string values to float by the help of age convertor fucntion which is define in my code and fill missing values with the rounded mean and mode. We perform one-hot encoding on selected columns, excluding 'Primary Diagnosis'. The 'Age' column is then scaled between 0 and 1 using MinMax scaling.

3.2 Feature selection

Now that we've made sure our data is in a consistent format, we want to decide which features are the most important for our predictions. In our dataset, we have 55 features after one hot encoder. While all of them could be essential, we need to experiment and check if our results improve by focusing only on the most crucial features suggested by different algorithms. During the training phase, we'll compare two scenarios: one using the original data with all features and the other using a trimmed-down version with only the most important features. By doing this, we aim to identify which approach gives us better results. Looking at the correlation matrix above, we observe that some features show little connection to the target variable. Therefore, we need to investigate whether removing these less influential features affects the overall performance, specifically in terms of the macro average and F1 score. This comparison will help us determine the impact of feature selection on the quality of our predictions.

3.3 Data Spliting for training and testing

Data splitting is a crucial step in machine learning, where the dataset is divided into training and testing sets. This ensures the model's effectiveness on new, unseen data. The goal is to assess how well the model generalizes, preventing overfitting to the training data and ensuring reliable predictions.

4 Training the Classifiers

As mentioned earlier we have used five Classifiers. Namely Random Forest, Decision Tree, Logistic Regression, Support Vector and KNN. We have used the hyper-parameter tuning for all the above mentioned 5 models.

4.1 K - Nearest neighbour

: We have used this Parameters: imputer strategy:mean, median, most frequent, knnclassifier n neighbors: [3, 5, 7,9,11], knn classifier weights:[uniform, distance] in grid search and we got the imputer strategy': 'mean', 'knn classifier n neighbors': 11, 'knn classifier weights': 'distance'

4.2 Support Vector Machine(SVM)

: We have used 'C': [0.1, 1, 10, 100], 'gamma':['scale', 'auto'], 'kernel': ['linear','poly','rbf','sigmoid'] These parameters in grid search cv to train model and we got 'C': 0.1, 'gamma': 'scale', 'kernel': 'linear'

4.3 Decision Tree

: We have used following parameter in grid search cv to train the model 'criterion': ['gini', 'entropy'], 'max depth': [3, 5, 6, 8, 10], 'min samples split': [2, 4, 6, 8, 10], 'min samples leaf': [1, 2, 3, 4, 5] and we got 'criterion': 'gini', 'max depth': None, 'min samples leaf': 1, 'min samples split': 2 as best parameter.

4.4 Random Forest

: We have used following parameter in grid search cv to train the model 'criterion': ['gini', 'entropy'], 'n estimators': [10, 5, 100], 'max depth': [None, 5, 10], 'min samples split': [2, 5, 10] and we got 'criterion': 'entropy', 'max depth': 10, 'min samples split': 2, 'n estimators': 100 , 'min samples leaf': 1,as best parameter.

4.5 Logistic Regression

: We have used the following parameter in grid search cv to train the model

imputer strategy: ['mean', 'median', 'most frequent'], pca n components: [2, 5, 10], logreg classifier C: [0.001, 0.01, 0.1, 1, 10, 100], logreg classifier penalty: ['l1', 'l2'], logreg classifier solver: ['liblinear'] out of these we got imputer strategy: ['mean'], logreg classifier C: 100, logreg classifier penalty: l2, logreg classifier solver: ['liblinear'], pca n components: 10 as best parameter.

5 Evaluation Criteria

We are dealing with a classification problem, and for such problems, key evaluation criteria include Precision, Recall, F1 Score, Macro Average, and Micro Average. In our case, we specifically focus on F1 Score and Macro Average.

F1 Score

The F1 Score is chosen as an evaluation criterion because it represents the harmonic mean of Precision and Recall, providing a balanced measure of a model's performance.

$$\text{F1 Measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Macro Average

Macro Average involves calculating the precision and recall scores for each class from the confusion matrix and then averaging these scores across all classes.

$$\text{Macro Averaged Precision} = \frac{1}{m}\sum_{i=1}^{m}\frac{T_{Pi}}{T_{Pi}+F_{Pi}} \quad (1)$$

$$\text{Macro Averaged Recall} = \frac{1}{m}\sum_{i=1}^{m}\frac{T_{Pi}}{T_{Pi}+F_{Ni}} \quad (2)$$

Here, m represents the number of classes, T_{Pi} denotes True Positives, F_{Pi} represents False Positives, and F_{Ni} denotes False Negatives.

6 Result

Here we have taken mean of value of F1 score of 2 class labels for each model We have done here the Feature Engineering.So first we have run whole models without dropping any columns and got the results as shown below. As we can see from above co relation matrix that some of the features are nearly independent to the target variables.. So we test the model by dropping Primary Diagnosis features . By dropping the features we tend to know that F1 score of most of the model decreases

But After dropping the feature we got results that value of f1 score and macro average are shown below

Classifier	F1 Score	Macro Average
SVM	.85	.86
KNN	0.79	0.78
RF (Random Forest)	0.84	0.83
LR (Logistic Regression)	0.87	0.85
DT (Decision Tree)	0.82	0.8

Table 1: Classifier Performance Metrics after excluding primary diagnosis from data

Classifier	F1 Score	Macro Average
SVM	1.0	1.0
KNN	0.92	0.91
RF (Random Forest)	1.0	1.0
LR (Logistic Regression)	0.96	0.97
DT (Decision Tree)	1.0	1.0

Table 2: Classifier Performance Metrics before excluding primary diagnosis

7 Discussions and Conclusion

Support Vector Machine and losistic regression model is so far the best performing classifier on dataset in which we excluded the primary diagnosis feature. So Support Vector machine as our final model.

We have predicted the class label of test data using support vector machine as our model. for given test dataset. Advantages of our model 1.Improved decision-making: A Brain Tumor prediction model can help in healthcare make more informed decisions about the predicting the brain tumor. 2.Increased efficiency: By predicting the Brain tumor of the pateints,Doctor can better manage their treatment strategies,prescriptions and optimize their effort . Disadvantages of our model 1.Complexity: Developing and implementing a brain tumor prediction model can be complex and require significant resources, including data, computing power, and expertise. Future Improvement Real-time updates: Brain Tumor prediction models can be made more useful by providing real-world Data .

8 Reference

•Tanmay Sir Class notes

•https://scikit-learn.org/stable/

•Tutorials of machine learning

9 Github Link

https://github.com/avinash064/MLProject.git