Analyzing and Benchmarking Popular Classifiers on Figshare Brain Tumor Dataset

Matta Varun (19CS30028)

Indian Institute of Technology, Kharagpur

Department of Computer Science and Engineering, IIT Kharagpur

Data Analytics (CS61061) Course Project

Autumn 2023

Project Code : DA-15

Abstract

This project focuses on the analysis of brain tumor images from figshare brain tumor dataset using machine learning techniques. This dataset comprises 3064 samples with corresponding labels and tumor masks. It is processed using a specialized class, BrainTumorDataset, and features are extracted using a pre-trained ResNet18 model. Three classifiers, Gaussian Naive Bayes, Decision Tree, and Support Vector Machine, are trained and evaluated on the extracted features. Results demonstrate the classifiers' performance, providing insights into their accuracy, precision, recall, and other metrics. The report outlines the entire process, from data preprocessing to classifier evaluation, offering a comprehensive view of the methodology and outcomes.

Keywords: Brain Tumor, Machine Learning, Feature Extraction, ResNet18, Gaussian Naive Bayes, Decision Tree, Support Vector Machine, Classification

Analyzing and Benchmarking Popular Classifiers on Figshare Brain Tumor Dataset

The goal of this project is to develop a brain tumor classification system using deep learning techniques. The project involves loading a dataset of brain tumor images (the figshare brain tumor dataset), preprocessing the data, using a convolutional neural network (CNN) like ResNet18 for extracting features, and evaluating the performance of different classifiers - Bayesian, Decision Tree and Support Vector Machines (SVM).

Methodology

Loading Dataset and Pre-processing

We begin by loading the dataset. The dataset is unconventional in the sense that the samples are as .mat files. The ***BrainTumorDataset*** class, made from the *torch.utils.data.dataset* class, is responsible for loading and processing the brain tumor dataset. It uses *h5py* to read .mat files. For each file, it loads the image, label, and tumor mask, applies preprocessing (such as resizing the image, etc., which we have analyzed to see the impact of preprocessing on the performance), and stores them in self.images and self.labels. We also utilize *torch.nn.function.interpolate* for resizing all images to the same size. It handles exceptions and prints an error message if any issues occur during data loading.

Then the dataset splitting happens. We split it in 80:10:10 ratio, representing training dataset, testing dataset and validation dataset. The code initializes an instance of *BrainTumorDataset* and prints the sizes of the training, validation, and test sets. It then creates DataLoader instances (trainloader, valloader, and testloader) for each set.

Extracting Features

A pretrained ResNet18 model is loaded, and its final fully connected layer is removed. This modified ResNet18 is then used to extract features from the training, validation, and test datasets. It flattens the output tensor to a flat vector and stacks the list of feature arrays vertically into a single NumPy array, which is returned, along with a concatenated list of label arrays in the form of a single NumPy array.

Model Evaluation Criteria

The *evaluate_classifier* function computes various classification metrics such as accuracy, precision, recall, F1 score, AUC- ROC score, and confusion matrix. It then prints these metrics for the specified classifier *model_name*.

Training and Evaluation of Classifiers

Three classifiers - Bayesian, Decision Tree and SVM, are trained and evaluated.

1. Bayesian Classifier (**GaussianNB**): The code initializes, fits, and predicts using a Gaussian Naive Bayes classifier from *sklearn.naive_bayes*. The predictions are then evaluated using the *evaluate_classifier* function.

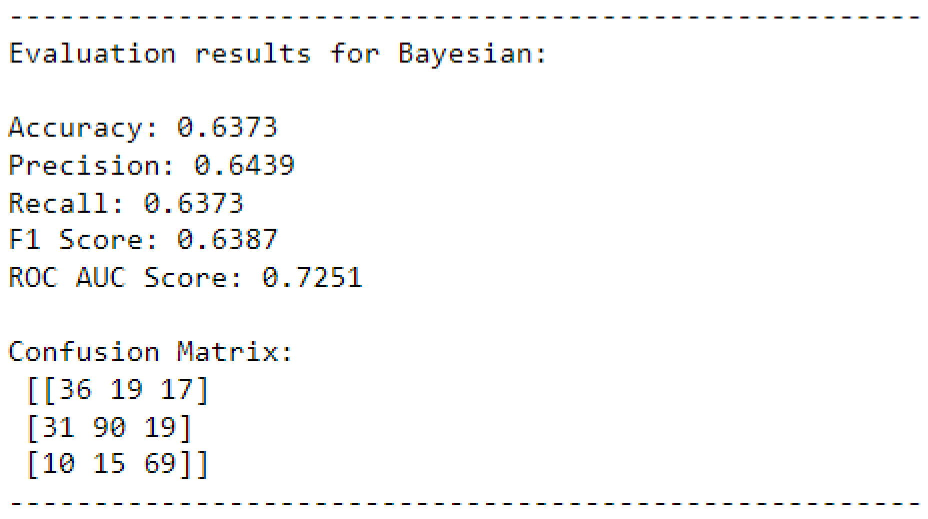
2. Decision Tree Classifier (**DecisionTreeClassifier**): Similar to the Bayesian classifier, a decision tree classifier from *sklearn.tree* is initialized, fitted, and predicted. The predictions are evaluated using the *evaluate_classifier* function.
3. SVM Classifier (**SVC**): A Support Vector Machine (SVM) classifier from *sklearn.svm* is initialized, fitted, and predicted. The predictions are evaluated using the *evaluate_classifier* function.

Results and Evaluation

The evaluation results, including accuracy, precision, recall, F1 score, ROC AUC score, and confusion matrix, are given below for each classifier..

Bayesian Classifier

With Tumor Masking

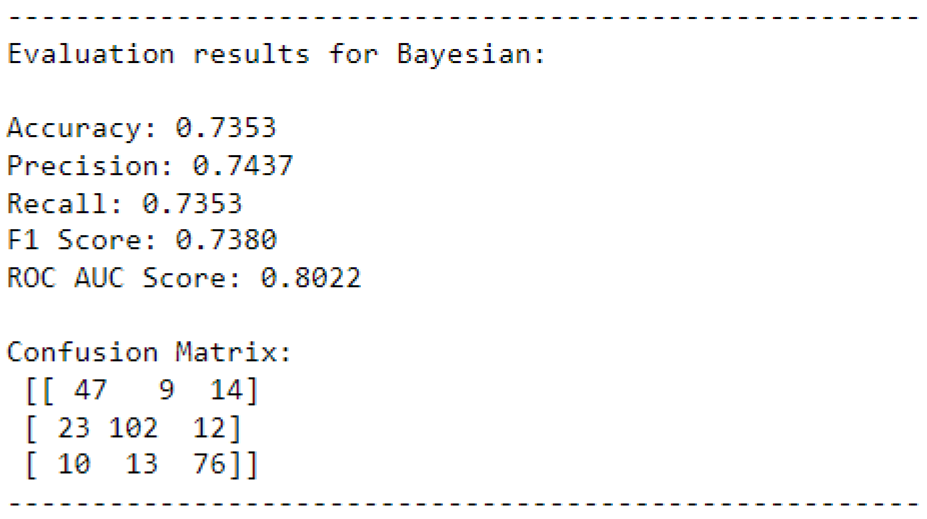


```
-----------------------------------------------------
Evaluation results for Bayesian:

Accuracy: 0.6373
Precision: 0.6439
Recall: 0.6373
F1 Score: 0.6387
ROC AUC Score: 0.7251

Confusion Matrix:
 [[36 19 17]
 [31 90 19]
 [10 15 69]]
-----------------------------------------------------
```

<u>Without Tumor Masking</u>



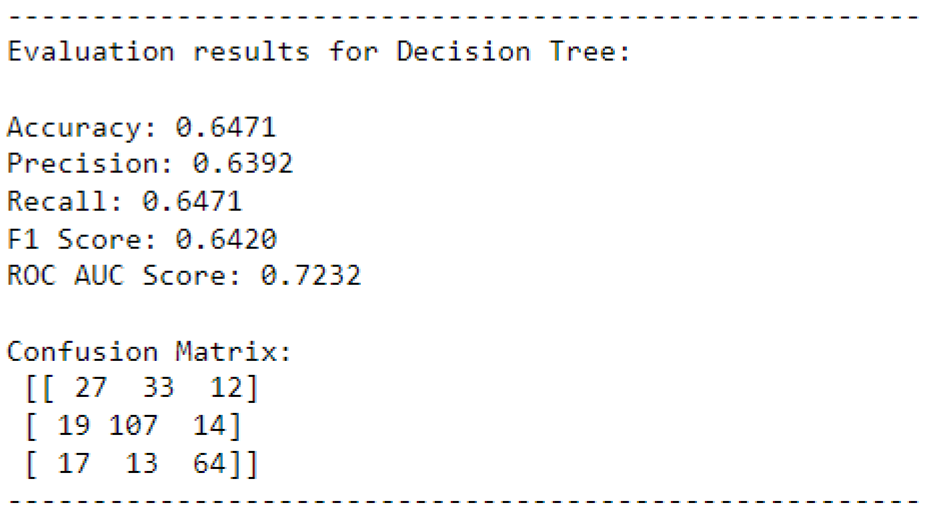
```
------------------------------------------------------
Evaluation results for Bayesian:

Accuracy: 0.7353
Precision: 0.7437
Recall: 0.7353
F1 Score: 0.7380
ROC AUC Score: 0.8022

Confusion Matrix:
 [[ 47   9  14]
 [ 23 102  12]
 [ 10  13  76]]
------------------------------------------------------
```

Decision Tree Classifier

<u>With Tumor Masking</u>

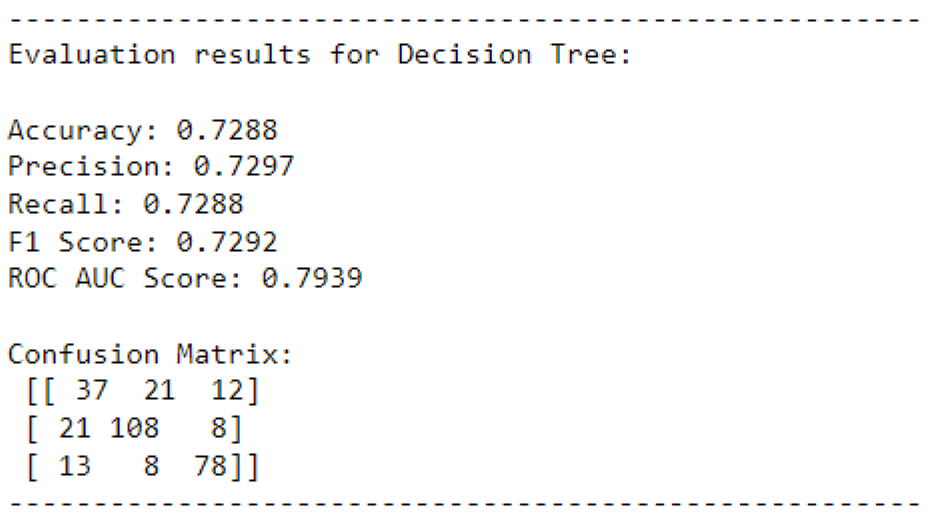


```
------------------------------------------------------
Evaluation results for Decision Tree:

Accuracy: 0.6471
Precision: 0.6392
Recall: 0.6471
F1 Score: 0.6420
ROC AUC Score: 0.7232

Confusion Matrix:
 [[ 27  33  12]
 [ 19 107  14]
 [ 17  13  64]]
------------------------------------------------------
```

<u>Without Tumor Masking</u>



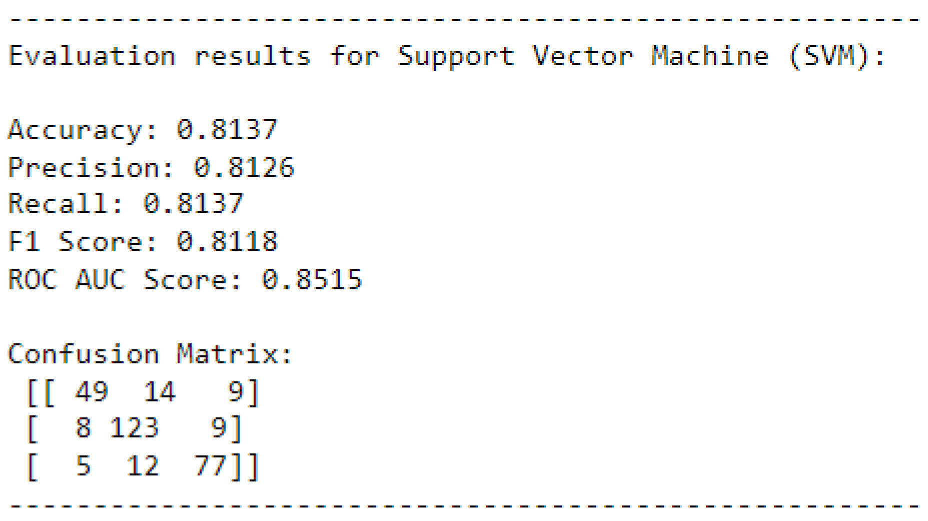
```
-------------------------------------------------------
Evaluation results for Decision Tree:

Accuracy: 0.7288
Precision: 0.7297
Recall: 0.7288
F1 Score: 0.7292
ROC AUC Score: 0.7939

Confusion Matrix:
 [[ 37  21  12]
 [ 21 108   8]
 [ 13   8  78]]
-------------------------------------------------------
```

SVM Classifier

<u>With Tumor Masking</u>

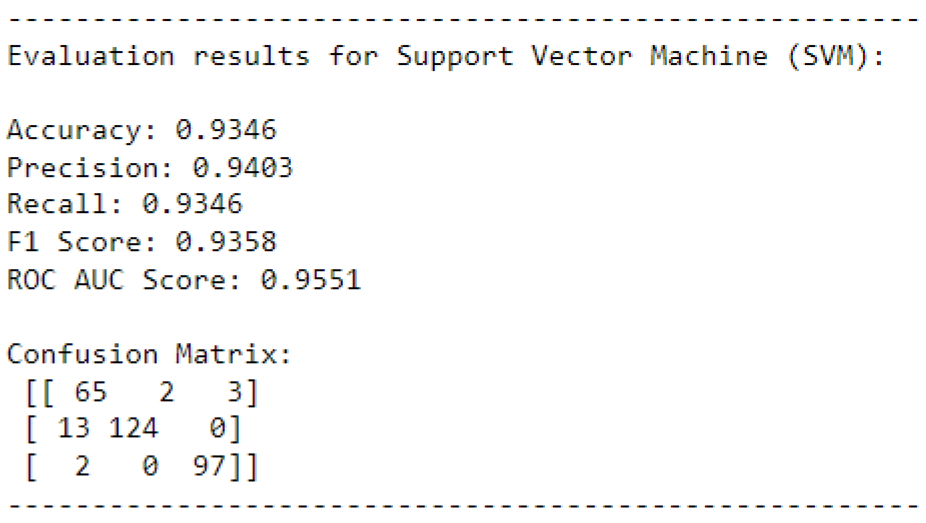


```
-------------------------------------------------------
Evaluation results for Support Vector Machine (SVM):

Accuracy: 0.8137
Precision: 0.8126
Recall: 0.8137
F1 Score: 0.8118
ROC AUC Score: 0.8515

Confusion Matrix:
 [[ 49  14   9]
 [  8 123   9]
 [  5  12  77]]
-------------------------------------------------------
```

Without Tumor Masking



```
------------------------------------------------------
Evaluation results for Support Vector Machine (SVM):

Accuracy: 0.9346
Precision: 0.9403
Recall: 0.9346
F1 Score: 0.9358
ROC AUC Score: 0.9551

Confusion Matrix:
 [[ 65   2   3]
 [ 13 124   0]
 [  2   0  97]]
------------------------------------------------------
```

Conclusion

The project successfully loads, preprocesses, and trains classifiers on the brain tumor dataset. From a performance perspective, SVM gives the best results, followed by Decision Tree and finally Bayesian Classifier. The evaluation results provided insights into the relative performance of these classifiers. Further improvements or modifications can be made based on these results.

THE END