Brain Tumor Classification Using Principal Component Analysis and Machine Learning

Shreya Savant
ID: 40243966
Github Link: https://github.com/shreyasavant/INSE_6220_Project

***Abstract*— Principal Component Analysis (PCA) is a statistical approach for reducing dimensionality in a dataset while maintaining the majority of the variability present in the original data. PCA assists in identifying and representing the most relevant patterns or features in a dataset in a more compact form. For this report, PCA is used to classify Brain tumors using three different classification algorithms, namely, Logistic Regression (LR), Random Forest (RF), and Quadratic Discriminant Analysis (QDA) on the original dataset and modified dataset (after PCA). The performance of each method is measured using the F1 score, confusion matrix, and receiver operating characteristic (ROC) curves after each model is tweaked using optimum hyperparameters to produce improved performance metrics. Additionally, decision boundaries for each model are displayed to demonstrate how well the model fits the dataset. RF outperforms all other existing machine-learning models in the PyCaret package. Finally, an analysis with Shapley values is performed to interpret the model. The extra trees (ET) classifier model is utilized for this. Overall, the algorithms successfully determine the two classes of brain tumor dataset and acquire an F1-score close to 1.**

Keywords—Principal Component Analysis, Machine Learning, Logistic Regression, Quadratic Discriminant Analysis, Random Forest

I. INTRODUCTION

Brain tumours are a significant health concern that affects millions of people worldwide. According to the American Brain Tumour Association, brain tumours account for approximately 85,000 diagnoses each year in the United States alone, with approximately 17,000 deaths attributed to these tumours [1]. Brain tumours are caused by abnormal growth of cells in the brain tissue, and these growths can be tumour or non-tumour.

Non-tumour brain tumours, also known as brain cancer, can rapidly spread to other areas of the brain and even to other parts of the body, making them particularly dangerous. In contrast, tumour tumours tend to grow slowly and are less likely to spread beyond the immediate area of the brain where they originated [2].

Early detection and diagnosis of brain tumours is essential for effective treatment and improved outcomes for patients. Machine learning and data mining techniques have shown promise in aiding the diagnosis process by identifying patterns and characteristics that can distinguish between tumour and non-tumour tumours [3]. These methods can help reduce the number of false positives and false negatives in brain tumour diagnosis, allowing for more accurate and timely treatment for patients.

Machine Learning techniques have evolved as a strong tool for identifying brain tumours in recent years, using classification algorithms to identify the type of tumour, forecast prognosis, and suggest the most appropriate treatment approach. Clinicians rely on accurate classification to deliver personalized care to their patients. In this paper, we use Principle Component Analysis (PCA) to reduce dimensionality in a brain tumour dataset.

Following that, three popular classification algorithms, namely Logistic Regression (LR), Random Forest Classifier (RF), and Quadratic Discriminant analysis (QDA), are applied to the original and PCA-transformed datasets in order to differentiate between tumours and non-tumours. The goal is to establish whether the tumour is present or not. Explainable AI (Artificial Intelligence) Shapley values are utilised to interpret the classification models. It is worth noting that the classification results presented in this report are obtained from the transformed dataset.

II. EXPLORATORY DATA ANALYSIS

A. Raw Data Set Description

The dataset contains features extracted from 3762 images from Brats2015 brain tumor with different slices of different MRI image [5] [6]. This is a brain tumor feature dataset including five first-order features and eight texture features with the target level (in the column Class). First Order Features are Mean, Variance, Standard Deviation, Skewness, Kurtosis and the Second Order Features are as follows Contrast, Energy, ASM (Angular second moment), Entropy, Homogeneity, Dissimilarity, Correlation, Coarseness. The Image column defines image name and Class column defines either the image has tumor or not. (1 = Tumor, 0 = non-Tumor).

B. Data Cleaning

The dataset had a column named Image that denoted the image name from which the data was extracted. This column was dropped during preprocessing. For the purpose of this study, only training data was used.

C. Description of Used Data Set

```
RangeIndex: 3762 entries, 0 to 3761
Data columns (total 14 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Image               3762 non-null   object
 1   Class               3762 non-null   int64
 2   Mean                3762 non-null   float64
 3   Variance            3762 non-null   float64
 4   Standard Deviation  3762 non-null   float64
 5   Entropy             3762 non-null   float64
 6   Skewness            3762 non-null   float64
 7   Kurtosis            3762 non-null   float64
 8   Contrast            3762 non-null   float64
 9   Energy              3762 non-null   float64
 10  ASM                 3762 non-null   float64
 11  Homogeneity         3762 non-null   float64
 12  Dissimilarity       3762 non-null   float64
 13  Correlation         3762 non-null   float64
dtypes: float64(12), int64(1), object(1)
memory usage: 411.6+ KB
```

Fig. 1: Description of Dataset

The description of the dataset is given in Fig. 1. The dataset contained no duplicate rows or NaN values. And all of the values are floating point numbers, with the exception of the "Class" column, which is designated with int values 0 and 1.

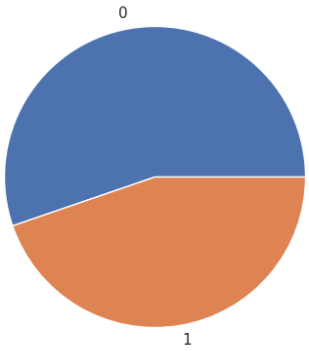


Fig. 2: Distribution of Class

Distribution of class is shown in Fig. 2. As mentioned, the label 1 denotes a tumor and Class 0 denotes non-tumor scan.

D. Data Analysis

Standardization is put into use to adjust each input variable independently by removing the mean, and dividing by the standard deviation to shift the distribution to have a mean of zero and a standard deviation of one [4].

Plotting the standardize matrix in a Box Plot gives us an idea of the data distribution, as well as measures of central tendency and spread. Fig. 3 shows that all of the data attributes are positively biased to some extent. Because of standardization, data is centered around 0 and fluctuation is kept to a minimum. Outliers are evident for four of the twelve traits, and they are all on the skewed side of whiskers.

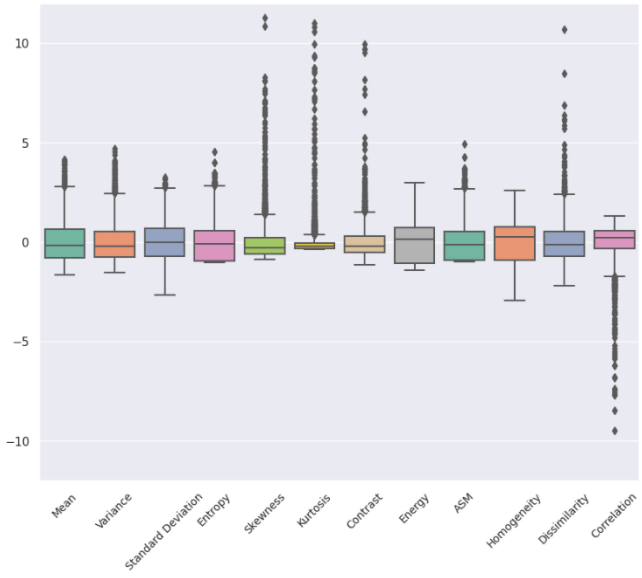


Fig. 3: Box Plot

Correlation Matrix and Pair Plot are used to understand the relationship between the qualities. Almost all of the factors are clearly positively connected with one another.

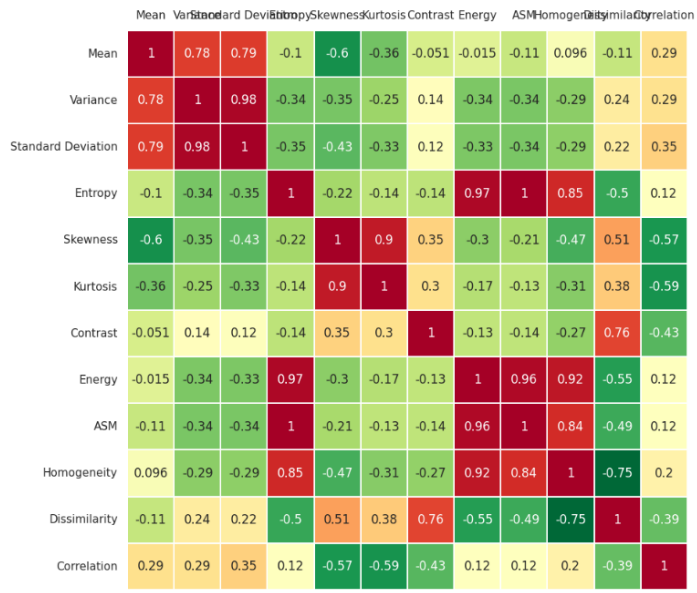


Fig. 4: Correlation Matrix

It is observed that there is negative correlation between some variables, while most are positively correlated. While Entropy and Contrast have significant correlation with the other components, Kurtosis has comparatively lesser.

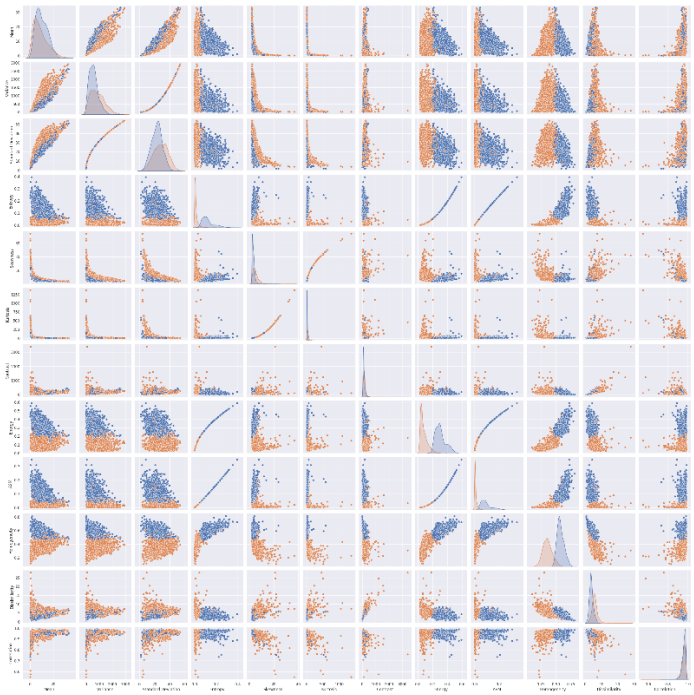


Fig. 5: Pair Plot

This is additionally supported by the Pair Plot in Fig. 5. The increasing trend in the line determines the strong positive associations. In the pair plot, weak correlations form clusters rather than an increasing line.

PCA is used to obtain un-correlated data because of the various correlations among data set attributes.

III. PRINCIPAL COMPONENT ANALYSIS

Most real-world datasets are exceedingly complicated, with a huge number of variables or features that make processing, storing, and visualizing the data difficult. This

complexity can result in significant computational costs and may even make effective data manipulation impossible.

Principal component analysis (PCA) is a method of extracting important variables (in the form of components) from a large set of variables available in a data set. It extracts low dimensional set of features from a high dimensional data set with a motive to capture as much information as possible. With fewer variables, visualization also becomes much more meaningful [4].

PCA is an effective approach for decreasing the complexity of high-dimensional data while preserving significant patterns and trends. PCA allows academics to more readily study and visualise data by condensing the data into a fewer number of dimensions. The resulting feature summaries depict the original data in a succinct yet complete manner, making it easier to analyse and work with. Overall, PCA is an important technique for dealing with high-dimensional data challenges in many real-world applications.

A. Principal Component Analysis Algorithm

To apply PCA to a data matrix X with dimension n x p, the following steps are typically taken [7]:

1) Standardization: The initial variables are standardized so that they all contribute equally to the analysis. This is done by computing the mean vector $\bar{x}$ of each column of the data set which is calculated as follows:

$$\bar{x} = \frac{1}{n}\sum_{n=1}^{n} x_i$$

By subtracting the mean of each column from each item in the data matrix, the data is standardized. The final centered data matrix (Y) is written as follows:

$$Y = H\,X$$

where H is the centering matrix.

2) *Covariance matrix computation:* This step's goal is to determine the relationship between the variables. When variables are closely connected, they can include redundant information. The covariance matrix is used to identify these associations. The covariance matrix p p is calculated as follows:

$$S = \frac{1}{n-1} Y^T Y$$

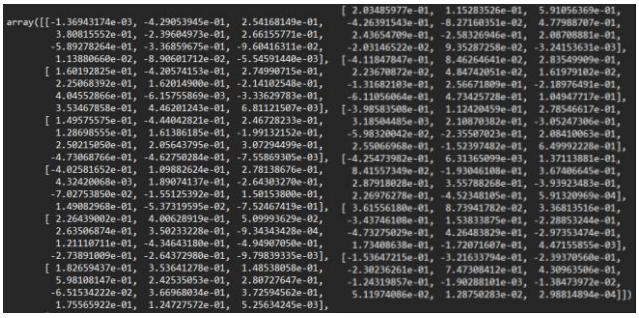


Fig. 6: Eigenvectors

```
Array
([4.90710566e+00,       3.75689637e+00,
1.54532242e+00,         7.51981437e-01,
5.74289358e-01,         2.40128855e-01,
1.29225878e-01,         4.01238990e-02,
3.06448213e-02,         1.67032410e-02,
1.07042171e-02,       6.44871526e-05])
```

Fig. 7: Eigenvalues

The first PC is given by:

$Z = 0.49071\ X_5 + 0.1545\ X_4 + 0.5742\ X_3 + \ldots + X_1$

3) Eigen decomposition: The eigenvalues and eigenvectors of S can be determined using the eigen decomposition. Eigenvectors show each principle component's (PC) direction, whereas eigenvalues represent the variance captured by each PC. The following equation can be used to calculate Eigen decomposition:

$$S = A\Lambda A^T$$

where A is the p x p orthogonal matrix of eigenvectors and Λ is the diagonal matrix of eigenvalues. The eigenvector matrix and the matrix of eigenvalues, both are given in Fig 6 & 7, respectively.

4) Principal components: It computes the converted matrix Z of size (n x p). The observations are represented by the rows of Z, while the PCs are represented by the columns of Z. The number of PCs is equal to the original data matrix's dimension. The equation for Z is given by:

$$Z = Y\,A$$

The variance of j^{th} PC is given as following:

$$l_j = \frac{\Lambda_j}{\sum_j^p \Lambda_j} \times 100\%, f\ or\ j = 1, \ldots., p$$

where Λ_j gives the variance of j^{th} PC.

Both Scree/Elbow plots can be used to get an idea of how many PCs are needed to represent the variance present in the data.

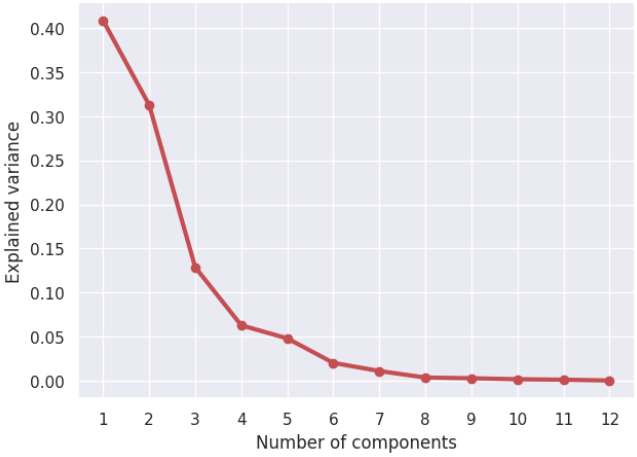


Fig. 8: Scree Plot

During the study, it was discovered that the variation accounted for by the first PC is l= 40.8%, the variance accounted for by the second PC is l= 31.2%, and the variance accounted for by 5 PCs is l= 98.10%. In the scree

plot, the elbow joint shows a bend at PC number 5, which is also supported by the Pareto Chart.

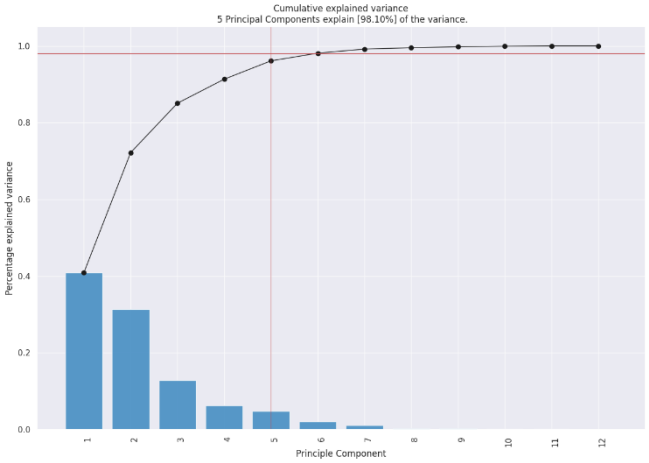


Fig. 9: Pareto Chart

As a result, it is safe to infer that the dimensions of eigenvector or Z components can be decreased to five.

This same could be verified with the help of PC coefficient Plot as in Fig.

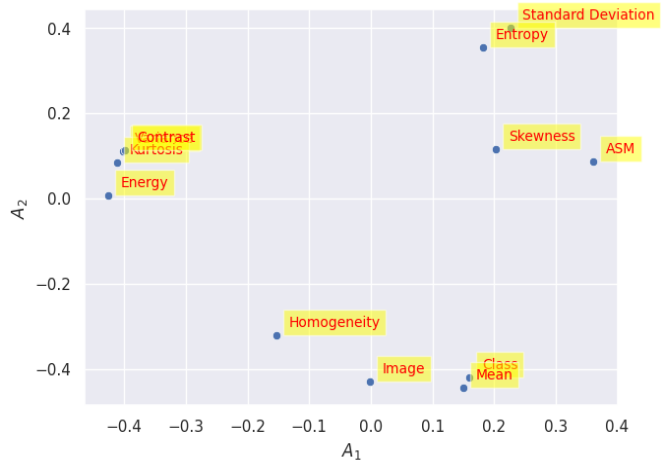


Fig. 10: PC Coefficient Plot

Contrast, Kurtosis, Energy, Variance lie in the lower range for the first PC, along with ASM, Skewness and Standard deviation being the most important factors for consideration, with Homogeneity and Mean being the middle two.

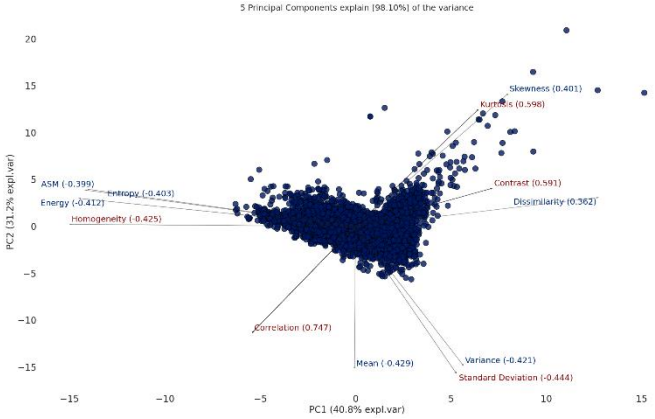


Fig. 11: Biplot

Biplot exhibits the same observations as Fig. 11. The angles between the vectors (rows of the eigenvector matrix) and the axes (representing the first two PCs) give the variable contribution to the PCs [3], i.e., the vector with the smallest angle with the axe contributes the most to that axe/PC. In addition, each observation is shown as a point on the plot.

This same is represented for 5 PCs with help of 3d Biplot in Fig. 12.

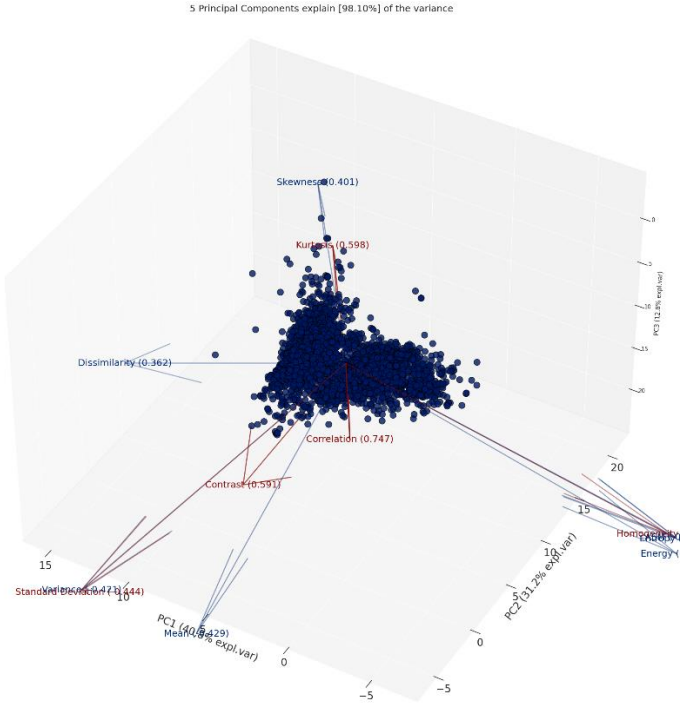


Fig. 12: 3D Biplot

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
xgboost	Extreme Gradient Boosting	0.9899	0.9981	0.9830	0.9944	0.9886	0.9795	0.9796	0.3490
lightgbm	Light Gradient Boosting Machine	0.9886	0.9983	0.9811	0.9934	0.9872	0.9769	0.9771	0.4860
ada	Ada Boost Classifier	0.9878	0.9973	0.9811	0.9915	0.9863	0.9752	0.9753	0.5920
rf	Random Forest Classifier	0.9873	0.9978	0.9764	0.9952	0.9857	0.9743	0.9746	0.5730
gbc	Gradient Boosting Classifier	0.9861	0.9980	0.9764	0.9925	0.9843	0.9718	0.9720	0.8160
et	Extra Trees Classifier	0.9861	0.9988	0.9764	0.9924	0.9843	0.9718	0.9720	0.5330
dt	Decision Tree Classifier	0.9831	0.9826	0.9774	0.9848	0.9810	0.9658	0.9660	0.2500
qda	Quadratic Discriminant Analysis	0.9759	0.9930	0.9660	0.9803	0.9729	0.9513	0.9517	0.1320
lda	Linear Discriminant Analysis	0.9717	0.9950	0.9415	0.9951	0.9675	0.9425	0.9437	0.2380
lr	Logistic Regression	0.9671	0.9941	0.9358	0.9902	0.9620	0.9330	0.9345	0.5000
ridge	Ridge Classifier	0.9667	0.0000	0.9274	0.9980	0.9612	0.9321	0.9342	0.0990
nb	Naive Bayes	0.9641	0.9902	0.9349	0.9843	0.9588	0.9271	0.9283	0.1810
knn	K Neighbors Classifier	0.8101	0.8774	0.7368	0.8221	0.7758	0.6121	0.6162	0.1410
svm	SVM - Linear Kernel	0.8034	0.0000	0.7585	0.8254	0.7424	0.5952	0.6276	0.1280
dummy	Dummy Classifier	0.5527	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.2200

Fig. 13: Comparison among classification models before applying PCA

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
et	Extra Trees Classifier	0.9793	0.9942	0.9623	0.9913	0.9765	0.9581	0.9585	0.6230
lightgbm	Light Gradient Boosting Machine	0.9793	0.9936	0.9660	0.9877	0.9766	0.9581	0.9585	0.4700
xgboost	Extreme Gradient Boosting	0.9785	0.9926	0.9660	0.9857	0.9757	0.9564	0.9566	0.3140
rf	Random Forest Classifier	0.9772	0.9927	0.9623	0.9866	0.9742	0.9538	0.9542	0.5710
knn	K Neighbors Classifier	0.9747	0.9868	0.9528	0.9902	0.9711	0.9486	0.9493	0.7350
lr	Logistic Regression	0.9743	0.9928	0.9557	0.9866	0.9707	0.9478	0.9484	0.3740
ridge	Ridge Classifier	0.9743	0.0000	0.9547	0.9875	0.9707	0.9478	0.9484	0.1810
lda	Linear Discriminant Analysis	0.9743	0.9926	0.9547	0.9875	0.9707	0.9478	0.9484	0.2580
svm	SVM - Linear Kernel	0.9722	0.0000	0.9651	0.9730	0.9688	0.9437	0.9440	0.4580
gbc	Gradient Boosting Classifier	0.9713	0.9925	0.9575	0.9783	0.9676	0.9419	0.9423	0.7030
ada	Ada Boost Classifier	0.9675	0.9896	0.9528	0.9743	0.9633	0.9342	0.9345	0.5920
dt	Decision Tree Classifier	0.9650	0.9639	0.9538	0.9677	0.9605	0.9290	0.9294	0.2890
qda	Quadratic Discriminant Analysis	0.9650	0.9900	0.9330	0.9882	0.9597	0.9288	0.9302	0.1920
nb	Naive Bayes	0.9633	0.9885	0.9491	0.9688	0.9587	0.9257	0.9261	0.3870
dummy	Dummy Classifier	0.5527	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.4450

Fig. 14: Comparison among classification models after applying PCA

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.9705	0.9920	0.9434	0.9901	0.9662	0.9400	0.9409
1	0.9831	0.9972	0.9623	1.0000	0.9808	0.9657	0.9663
2	0.9705	0.9945	0.9434	0.9901	0.9662	0.9400	0.9409
3	0.9578	0.9815	0.9151	0.9898	0.9510	0.9140	0.9162
4	0.9789	0.9829	0.9623	0.9903	0.9761	0.9572	0.9575
5	0.9747	0.9975	0.9811	0.9630	0.9720	0.9489	0.9490
6	0.9747	0.9922	0.9623	0.9808	0.9714	0.9487	0.9488
7	0.9831	0.9992	0.9717	0.9904	0.9810	0.9658	0.9659
8	0.9705	0.9926	0.9623	0.9714	0.9668	0.9402	0.9402
9	0.9789	0.9985	0.9528	1.0000	0.9758	0.9571	0.9580
Mean	0.9743	0.9928	0.9557	0.9866	0.9707	0.9478	0.9484
Std	0.0072	0.0059	0.0174	0.0112	0.0084	0.0146	0.0142

Fig. 15: LR metrics score after hyperparameter tuning

IV. Machine Learning-Based Classification Algorithms

Classification involves assigning a label or category to a given input and determining whether a specific type belongs to a particular class. This technique can be applied to both structured and unstructured data and is considered a type of supervised learning [10]. Depending on the data being analysed, classification can involve binary or multi-class classification. Classification algorithms operate in various ways, such as logic-based based, Perceptron based, Statistic techniques, Support Vector Machines, and others.

Selecting one of these options is inconclusive as it relies heavily on the usage and characteristics of the data [11]. Therefore, to simplify the process for this study, PyCaret was used to make the choice, as depicted in Figure 13.

The Receiver Operating Characteristics (ROC) curve is commonly used to visually compare classification models by showing the relationship between the true positive rate and the false positive rate, with the area under the curve representing the model's accuracy [10]. In this study, the training data was split into a 70-30 ratio for testing purposes.

As shown in Figure 16, the Extra Trees Classifier performed the best on the original data without tuning. Three other models were selected based on their evaluation scores being similar to the best model: Decision Trees Classifier, Random Forest Classifier, and Light Gradient Boosting Machine Classifier. The tuning process was facilitated by PyCaret, and no explicit hyperparameter selection was performed for optimal model performance in this report [12].

A. *Logistic Regresssion*

Logistic regression is a statistical method used to analyse the relationship between a binary dependent variable and one or more independent variables. It is widely used in many fields such as finance, medicine, and social sciences. The goal of logistic regression is to estimate the probability of an event occurring based on a set of predictors [14].

The logistic regression model uses a logistic function to model the relationship between the independent variables and the dependent variable. The logistic function, also known as the sigmoid function, maps any real-valued input to an output between 0 and 1, which can be interpreted as a probability. The output of the logistic regression model is the predicted probability of the event occurring given the values of the independent variables.

$$S(z) = \frac{1}{1 + e^{-z}},$$

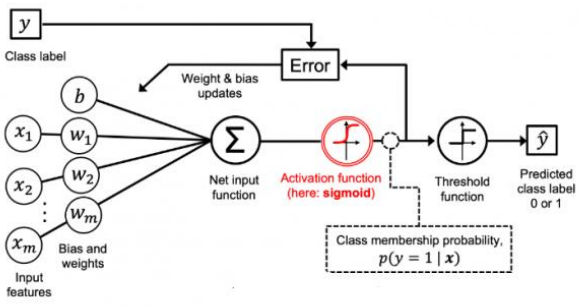


Fig. 14: Logistic Regression [15]

B. *Random Forest Classifier*

Random is a type of ensemble learning method that combines multiple decision trees to make predictions.

In a random forest, a large number of decision trees are trained on different subsets of the data, and the final prediction is made by averaging the predictions of all the individual trees. This helps to reduce overfitting and increase the accuracy and robustness of the model.

Random forest algorithm is widely used in various fields such as finance, healthcare, and marketing, and is particularly useful for handling large and complex datasets [16].

$$P = \frac{e^{a+bX}}{1 + e^{a+bX}}$$

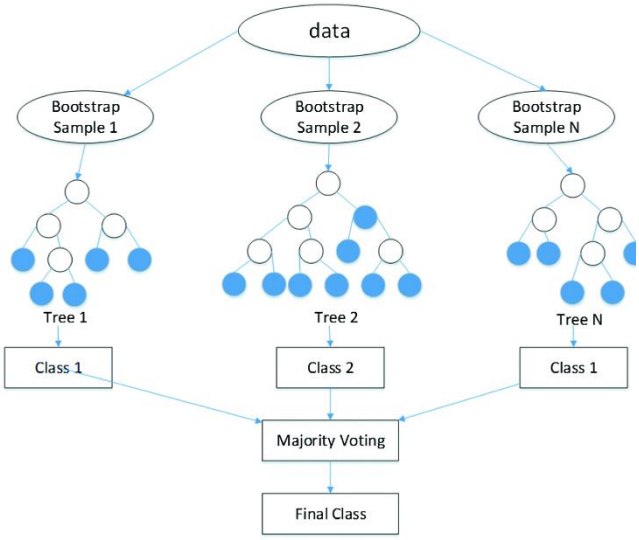


Fig. 15: Random Forest Classifier [17]

C. *Quadratic Discriminant Analysis*

Quadratic Discriminant Analysis (QDA) models each class as a multivariate Gaussian distribution and estimates the mean vector and covariance matrix for each class. Based on these estimates, QDA computes the discriminant function for each class, which is a quadratic function of the input features. The decision boundary between two classes is then the set of points where the two discriminant functions are equal [18].

$$\delta_y(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) + \log_{\pi_k}$$

QDA can be a powerful tool for classification tasks where the classes have complex, non-linear decision boundaries. However, it can be sensitive to overfitting when the number of features is large relative to the number of training samples.

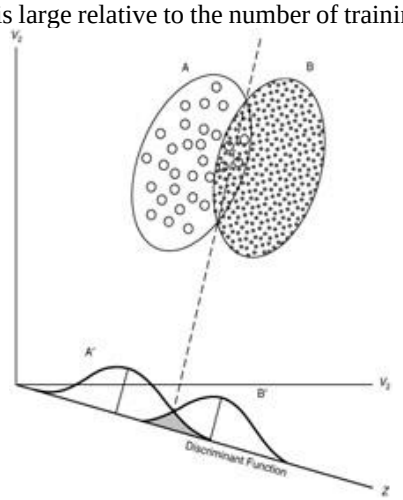


Fig. 16: Quadratic Discriminant Analysis [19]

Random forests are a method that can be used to reduce the variance and overfitting that can occur when using deep decision trees. In this approach, numerous decision trees are trained on different portions of the same training set, and their results are averaged to create a final model with improved performance [20][21]. Specifically, the classification rule for QDA can be described as below:

$$G(x) = arg\max_k \delta_k(x).$$

Since the number of QDA parameters is quadratic, QDA should be used with care when the feature space is large. While this method may result in a slight increase in bias and a loss of interpretability, it helps to address the issue of high variance associated with deep decision trees. The hyperparameters for random forests are similar to those used for decision trees and bagging classifiers, but with added randomness introduced during the model-building process [20]. By selecting features from a random collection of qualities, random forests can capture a wide range of variability and improve the overall accuracy of the model.

The trained and tuned model resulted in the ROC curve shown in Fig 20 with an AUC = 98.99% which was 97.93% without tuning of the model.

V. Classification + PCA Results

Three popular classification algorithms were applied to the brain tumour dataset to observe the effects of PCA on the dataset. The original dataset and the PCA applied dataset with three PCA components were used. The PyCaret library of Python was used to perform the classification, and the original dataset was split into a 70% train set and a 30% test set.

The session id was set to 123 for reproducibility purposes. By using PyCaret, a performance comparison table can be created among all available classification algorithms on the target dataset, and the best model with the highest accuracy can be found. Before applying PCA, the three best classification models with the highest accuracies on the brain tumour dataset were observed to be Extra Trees Classifier (ET), Light Gradient Boosting Machine (LightGBM), and Extreme Gradient Boosting (XGBoost).

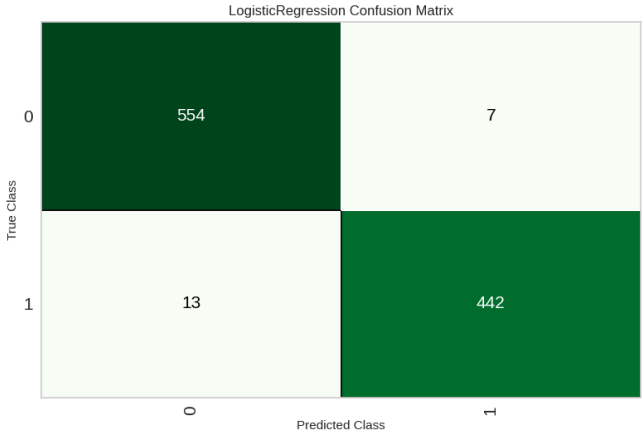


Fig. 17: Confusion Matrix for Logistic Regression

A ROC curve is a graph that shows how well a classification model performs across all categorization levels. This graph depicts two parameters: True Positive Rate and False Positive Rate. These are the primary components of the building confusion matrix. As a result, the ROC curve and the confusion matrix are closely related and can be viewed as distinct visual representations of the same data. The Google Colab notebook contains ROC curves for RF and QDA. The ROC curve of LR in Fig. 13 represents the confusion matrix results. It plots the false positive rate (x-axis) versus the genuine positive rate (y-axis) for a variety of candidate

threshold values ranging from 0.0 to 1.0. It also displays a graph of the macro and micro average curves.

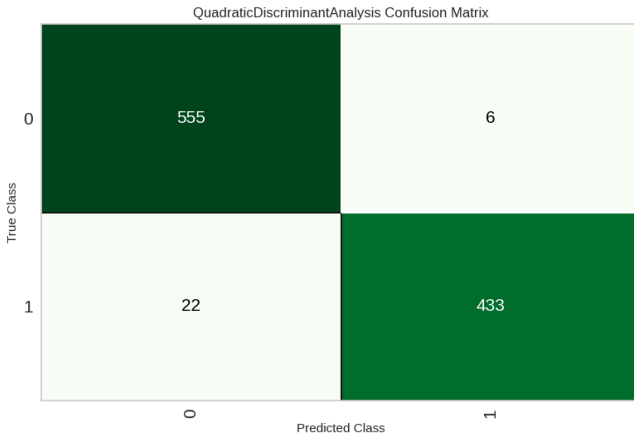


Fig. 18: Confusion Matrix for Quadratic Discriminant Analysis

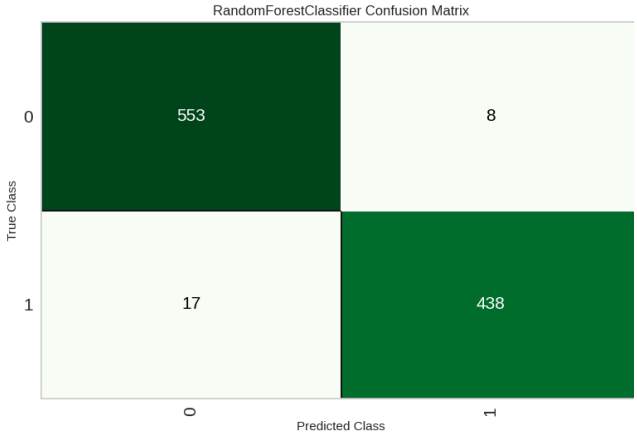


Fig. 19: Confusion Matrix for Random Forest Classifier

In this study, the performance of three classification algorithms - Logistic Regression (LR), Quadratic Discriminant Analysis (QDA), and Random Forest (RF) - was compared for the analysis of brain tumours using principal component analysis (PCA).

After applying PCA, it was observed that LR, Random Forest (RF), and QDA showed the highest accuracy on the transformed dataset. Therefore, these three algorithms were chosen for evaluation purposes. The original and transformed datasets were trained, tuned, and evaluated using these three algorithms. Hyperparameter tuning was performed using the PyCaret library, which involves creating a model, tuning it, and evaluating its performance.

The LR model was tuned using L2 penalty to prevent overfitting, while the number of RF and the reg_parameter for regularization were tuned for QDA. The metrics of the tuned LR model were better than those of the base model.

The decision boundaries formed by the algorithms on the transformed dataset were plotted, and it was observed that LR had the best decision boundary compared to RF and QDA. Precision and recall were used to evaluate the performance of each class individually, and the confusion matrices for the three algorithms were presented. LR outperformed all other models, including RF and QDA, with the lowest number of misclassifications.

F1-score, which combines precision and recall into a single metric, was used to compare the results among the classifiers. It was found to be a great metric for determining the better classifier in situations where one classifier has a higher recall and another has higher precision. The function of F1- score can be defined as below:

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall}.$$

Figures 18 and 19 indicate a significant improvement in the F1-score of LR, RF, and QDA after PCA is applied, suggesting that dimension reduction reduces the dependencies among the features. Additionally, LR and K-NN show further improvement in their F1-score after tuning with their ideal hyperparameters. These results highlight the benefits of PCA and hyperparameter tuning for brain tumour analysis using LR, RF, and QDA.

As a final analysis step, Fig. 13 shows the receiver operating characteristic (ROC) curve for the LR algorithm. A ROC curve is a graphical representation of a classification model's performance at all classification thresholds, plotting the True Positive Rate against the False Positive Rate. These parameters are key components in constructing a confusion matrix.

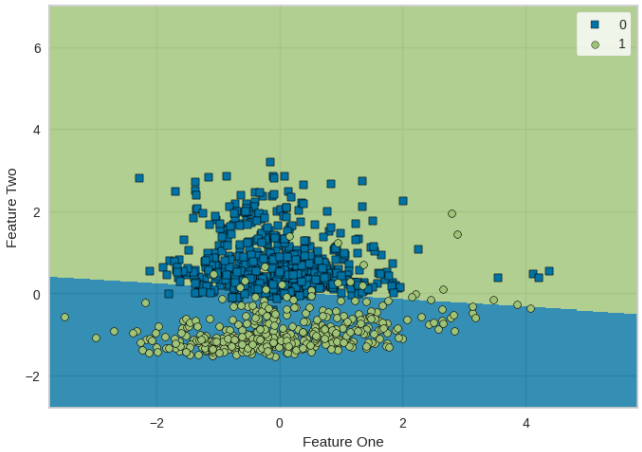


Fig. 20: Decision Boundary for Logistic Regression

As a result, the ROC curve and the confusion matrix are closely related and can be viewed as distinct visual representations of the same data. The Google Colab notebook contains ROC curves for RF and QDA. The ROC curve of LR represents the confusion matrix results. It plots the false positive rate (x-axis) versus the genuine positive rate (y-axis) for a variety of candidate threshold values ranging from 0.0 to 1.0. It also displays a graph of the macro and micro average curves. The ROC curve and AUC values demonstrate that LR is the best at predicting both classes, with an accuracy of 98%. This result is consistent with the confusion matrix results. As a result, the three algorithms are capable of correctly identifying the tumour and classes.

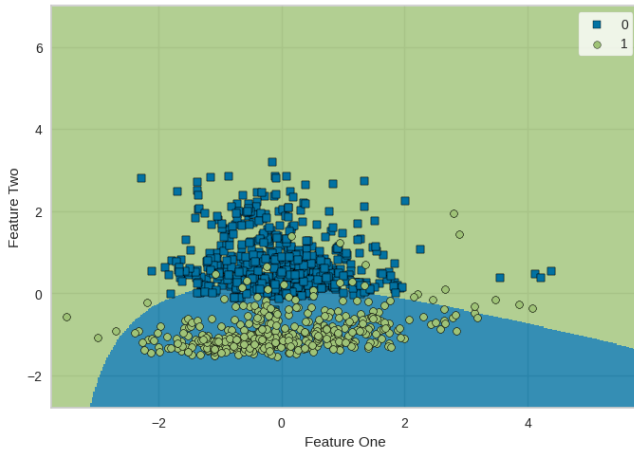


Fig. 21: Decision Boundary for Quadratic Discriminant Analysis

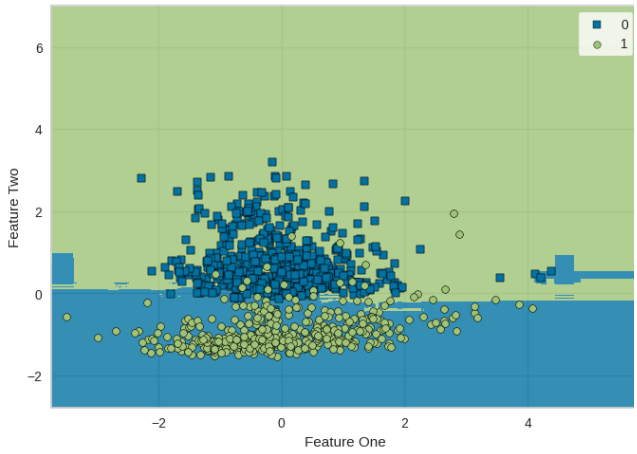


Fig. 22: Decision Boundary for Random Forest Classifier

VI. EXPLAINABLE AI WITH SHAPLEY VALUES

Model interpretability is a critical aspect of machine learning (ML) that can be enhanced using various methods, including feature importance. Feature importance estimates the contribution of each feature to the prediction process. To obtain an overview of the most important features on the PCs, the SHAP values are utilized by importing the open-source "SHAP" library of Python. SHAP (Shapley Additive Explanations) is a method that explains individual predictions based on optimal Shapley values using the concept of game theory. SHAP can explain the prediction of an instance x by computing the contribution of each feature to the prediction.

Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.9747	0.9888	0.9623	0.9808	0.9714	0.9487	0.9488
1	0.9916	0.9927	0.9811	1.0000	0.9905	0.9829	0.9830
2	0.9831	0.9911	0.9623	1.0000	0.9808	0.9657	0.9663
3	0.9705	0.9849	0.9340	1.0000	0.9659	0.9399	0.9416
4	0.9705	0.9901	0.9623	0.9714	0.9668	0.9402	0.9402
5	0.9789	0.9977	0.9811	0.9720	0.9765	0.9574	0.9574
6	0.9662	0.9971	0.9528	0.9712	0.9619	0.9316	0.9317
7	0.9831	0.9996	0.9717	0.9904	0.9810	0.9658	0.9659
8	0.9747	0.9904	0.9528	0.9902	0.9712	0.9486	0.9492
9	0.9789	0.9942	0.9623	0.9903	0.9761	0.9572	0.9575
Mean	0.9772	0.9927	0.9623	0.9866	0.9742	0.9538	0.9542
Std	0.0071	0.0043	0.0133	0.0114	0.0081	0.0144	0.0143

Fig. 23: RF before tuning

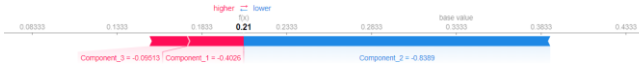
Fold	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.9662	0.9782	0.9434	0.9804	0.9615	0.9315	0.9320
1	0.9662	0.9928	0.9717	0.9537	0.9626	0.9319	0.9320
2	0.9747	0.9718	0.9528	0.9902	0.9712	0.9486	0.9492
3	0.9662	0.9870	0.9245	1.0000	0.9608	0.9312	0.9334
4	0.9662	0.9734	0.9528	0.9712	0.9619	0.9316	0.9317
5	0.9789	0.9905	0.9811	0.9720	0.9765	0.9574	0.9574
6	0.9705	0.9828	0.9717	0.9626	0.9671	0.9403	0.9404
7	0.9831	0.9952	0.9811	0.9811	0.9811	0.9659	0.9659
8	0.9662	0.9900	0.9528	0.9712	0.9619	0.9316	0.9317
9	0.9831	0.9852	0.9717	0.9904	0.9810	0.9658	0.9659
Mean	0.9722	0.9847	0.9604	0.9773	0.9686	0.9436	0.9440
Std	0.0069	0.0076	0.0173	0.0132	0.0079	0.0139	0.0137

Fig. 24: RF after Tuning

In the SHAP analysis, each feature of a dataset acts as a player in a coalition, where each player can be an individual feature value for tabular data or a group of feature values. Shapley values describe how to distribute the prediction adequately among the feature set. However, the SHAP library of Python is still in its development stage and only supports tree-based models, i.e., decision tree, random forest, and extra trees classifier, for binary classification problems. Since the Brain Tumour diagnosis dataset is a binary classification problem, SHAP analysis cannot be performed on LR, RF, and QDA. Therefore, for SHAP analysis, the fourth-best model of the transformed dataset, i.e., "Extra trees classifier (ET)," is chosen.

After creating and tuning the ET model with ideal hyperparameters, the tuned model is passed to the SHAP library to produce the interpretation plots. In this case, each PC acts as a player in the coalition. The summary plot of SHAP values is displayed in Fig. 14, which combines feature importance with feature effects. Each point on the summary

plot is a Shapley value for a PC and an instance. The y-axis represents the PCs, and Shapley values are positioned on the x-axis. All the PCs are ordered according to their importance.



The observation of the jittered overlapping points in the direction of the y-axis indicates the distribution of the Shapley values per PC. This observation supports the Pareto plot and scree plot, which indicate that the first PC holds the most feature variance. The red colour represents a high PC value, and the blue colour indicates a low PC value. To interpret the summary plot, a low level of PC value has a high and positive impact on the Brain Tumour diagnosis, while a high level of PC value has a low and negative impact on the Brain Tumour diagnosis. In other words, PCs are negatively correlated with the target variable.

VII. CONCLUSION

In conclusion, Principal Component Analysis (PCA) and three popular classification algorithms, namely Logistic Regression (LR), Quadratic Discriminant Analysis (QDA), and Random Forest (RF), are applied on the brain tumour dataset for brain tumour analysis. The dataset holds information on attributes of tumours to identify them as tumour or non-tumour. At first, PCA is applied on the original dataset. The first two Principal Components (PCs) apprehends 83% variance of the data. Hence, the feature set is reduced to 2 from 5. Extensive experiments are conducted on the first two PCs and different plots are generated to validate the obtained results from different perspectives.

To move forward, three classification algorithms (LR, RF, and QDA) are applied on the original dataset as well as transformed dataset with first three components. Each algorithm is tuned with the ideal hyperparameter settings, and performance evaluation is conducted by comparing confusion matrices, Receiver Operating Characteristic (ROC) curves, and F1-scores. It is observed that after hyperparameter tuning, the performance metrics score of each algorithm has improved significantly. The LDA, ET, and GBC algorithms performed the best on the original dataset. Interestingly, after applying PCA, LR, RF, and QDA performed the highest and showed the best performance metrics.

Finally, in order to increase the interpretability of the model, several interpretation plots are produced using explainable AI Shapley values. To summarize, all three algorithms (LR, QDA, and RF) can successfully determine the tumour types for brain tumour diagnosis.

REFERENCES

[1] American Brain Tumor Association. Brain Tumor Facts & Figures. (2021). https://www.abta.org/wp-content/uploads/2021/03/2021-ABTA-Facts-and-Figures.pdf

[2] National Institute of Neurological Disorders and Stroke. Brain Tumors Information Page. (2022). https://www.ninds.nih.gov/Disorders/All-Disorders/Brain-Tumors-Information-Page

[3] Wang, S., Liu, Z., Rong, Y., & Zhou, B. (2020). A machine learning-based framework for diagnosis of brain tumors using MR imaging. Computerized Medical Imaging and Graphics, 79, 101677. https://doi.org/10.1016/j.compmedimag.2019.101677

[4] Jolliffe, I. T. (2002). Principal component analysis. Wiley StatsRef: Statistics Reference Online. https://onlinelibrary.wiley.com/doi/book/10.1002/0471667196

[5] Menze et al., The Multimodal Brain TumorImage Segmentation Benchmark (BRATS), IEEE Trans. Med. Imaging, 2015.

[6] Kistler et. al, The virtual skeleton database: an open access repository for biomedical research and collaboration. JMIR, 2013.

[7] Jolliffe, I.T. Principal Component Analysis. 2nd ed. Springer, 2002.

[8] A. Ben Hamza, Advanced Statistical Approaches to Quality, unpublished.

[9] Bishop, C. M. (2006). Pattern recognition and machine learning (Vol. 4). New York: springer.

[10] Alpaydin, E. (2010). Introduction to machine learning (2nd ed.). Cambridge, MA: MIT Press.

[11] Kotsiantis, S. B. (2007). Supervised Machine Learning: A Review of Classification Techniques. Informatica, 31, 249-268.

[12] Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861-874.

[13] [12] PyCaret Team. (2022). PyCaret 2.3.4 documentation. Retrieved from https://pycaret.org/

[14] Hosmer Jr, D.W. and Lemeshow, S. (2004). Applied Logistic Regression (2nd ed.). John Wiley & Sons.

[15] https://vitalflux.com/python-train-model-logistic-regression/

[16] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[17] https://medium.com/@pranav3nov/understanding-random-forest-in-machine-learning-6db4daf74d19

[18] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.). Springer.

[19] https://uc-r.github.io/discriminant_analysis

[20] Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. R news, 2(3), 18-22.

[21] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.