**Models and classification**

**Using machine learning**

After all the encoding and cleaning of the data, we have encoded the data of DNA sequences and subpopulations, which are considered target variables in our case. We performed various multiclass-classification techniques to classify the subpopulation of the Oryza sativa species given.

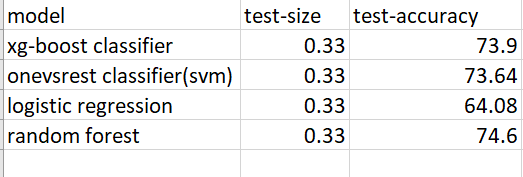
The first step before applying any predictive technique is to split the data into training and test data. The test size of all the models in our case is 0.33, i.e., 33% of the whole data. We have applied various classification models like XGBoost classifier, logistic regression (multinomial), random forest classifier, one-vs-one and one-vs-rest classifiers using the SVC kernel. Out of all the models, the random forest classifier and xgboost classifier gave descent accuracy.

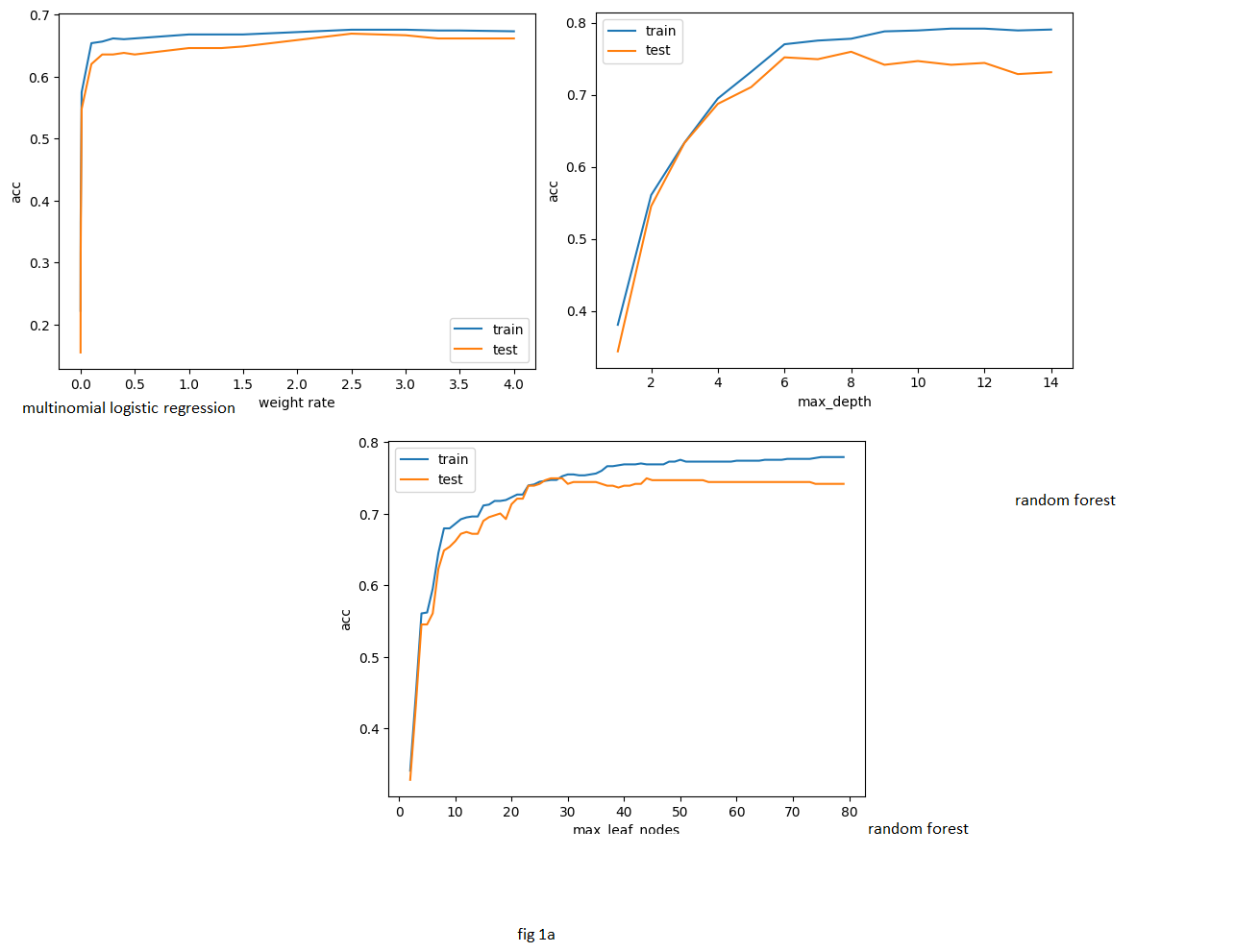
**Training and testing**

We trained our random forest model on the trained data and tuned the parameters of the classifier in order to prevent overfitting, and we also did the same for all other classifiers, which are shown in fig 1a. Selecting the number of trees and depth is important for random forest to prevent overfitting, despite random forest is less prone to overfitting. We can also use GirdSearchCv to find the best parameters of any model but this could be computationally expensive for large data sets. Although gridsearchcv finds parameters that prevent overfitting it is encouraged to tune the parameters by our own.

For random forest the accuracy for test data is 75 which isn’t a high accuracy but it’s a robust model which neither overfit nor underfit for the given data. Getting a high accuracy doesn’t matter if it overfit the data where model might not find better results for the new unseen data.

**Evaluating** the models is crucial in order to select the best among the trained models. We performed repeatedKStratified which is k-cross validation technique. It divides the data set into k splits and take one split as test data each time. This prevents the overfit and gives the actual accuracy of the model which considers the whole data and hidden relations in it.



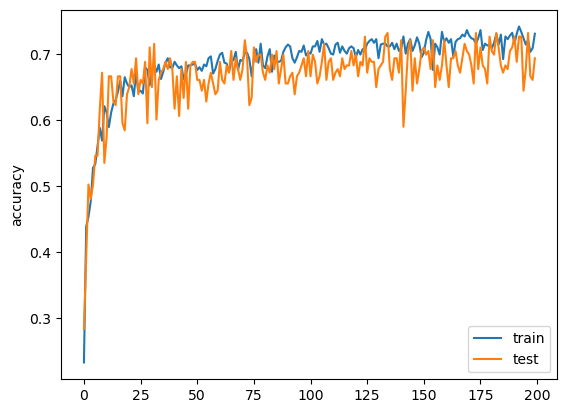


**Classification using ANN**

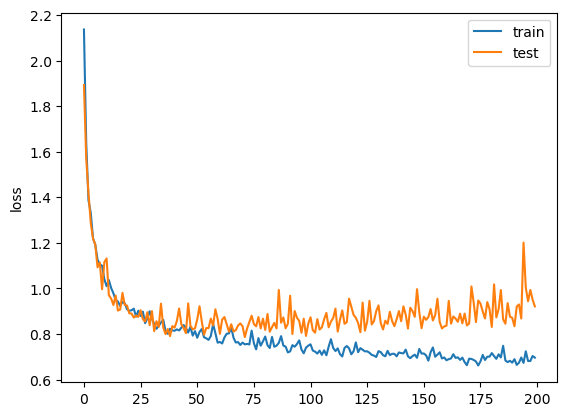
We used deep learning intended to capture the complex relations in the DNA sequences. But the truth is the neural networks require the large amount of data for data so it could give the better results than the traditional machine learning algorithms.

However, we created a simple multi-layer perceptron with just 2 hidden layers where the neural architecture is (9,16,16,10) where 10 is number of neurons (number of classes) in last layer with softmax activation function and remaining layers contain relu activation function. Kernel initializers like he\_normal and glorot\_uniform being used according to the layers. Finding the neurons and parameters is crucial and will have an impact on accuracy of the model. We just tried different number of neurons and parameters and selected best one. We can also use RandomizedSearchCv to find the params but could take a lot of time.

The model’s performance based on the number of epochs is shown in the Fig2a and Fig2b.



**Fig2a**



**Fig2b**

**Height prediction using Deep learning:**

**We utilized deep learning to predict the height of Oryza sativa (rice). However, traditional machine learning techniques yielded superior results compared to deep learning, primarily due to the requirement of large datasets for neural networks. Additionally, neural networks are prone to overfitting, wherein they become excessively specialized to the training set. To decrease overfitting, we employed dropout regularization and determined the optimal parameters using RandomizedSearchCV. As a result, we obtained a mean absolute error of approximately 17cm, which falls short of the performance achieved by traditional machine learning algorithms.**