Iris Dataset Report

Executive Summary

This study proposes two machine learning algorithms – Random Forest and Artificial Neural Network (ANN) to effectively perform multi-class classification on a dataset of kernel seeds. These experiments were conducted using Python v3.9 on Jupyter Notebook. The aforementioned models were evaluated through a Confusion Matrix, Accuracy, Precision, Recall, and F1-scores, whereby the mean accuracy scores of the best model of each algorithm were 95.2% and 93.8%, respectively.

Research Methodology

This study attempts to investigate the efficacy of Rain Forest and Artificial Neural Network models in multi-class classification of seeds. Three (3) models with varying configurations are conducted for each algorithm. To generate fairer and more valid results, five (5) iterations per model are performed which sums the total number of experiments to thirty (30).

As prerequisites to conduct this experiment, the following Python libraries were installed and initiated: pandas, numpy, matplotlib, and sklearn. The entire research process adopted a systematic approach comprised of seven (7) stages as illustrated in figure 5. The initial stage is to load the seeds dataset using pandas library and perform initial exploratory analysis. This is essential to do as it gathers fundamental information about the dataset, i.e., which are the predictor variables and which is the target variable, how many classes does the target have, are there any irrelevant columns and/or null values that would need to be cleaned? Once that is determined, the following step is to perform the actual cleaning of the dataset whereby three (3) redundant columns and several null values were deleted from the dataset. Additionally, seven (7) predictor variables and one (1) target variable were distinctly separated and labelled as x and y respectively.

Once all the preliminaries are completed, the following step is to conduct further exploration of all the variables to potentially gather more comprehensive insights about each of them. To that end, a series of boxplots and histograms were constructed and upon further inspection of each plot it could be observed that several predictor variables have overlapping distributions of the three (3) target classes, including outliers. This suggests that the nature

of the dataset may hinder the models' ability to classify the target classes correctly which may as a result manifest in the models' evaluation results.

The following stage is to transform the data accordingly to be able to be effectively processed in the Random Forest and ANN models. Firstly, a Principal Component Analysis (PCA) is conducted on the predictor variables, x. This technique reduces the dimensionality of a dataset to promote interpretability and improve processing performance while minimising information loss from the predictor variables. This is accomplished through the creation of new uncorrelated variables that successively maximise variance. To guarantee a correct PCA results, data standardisation is applied to set all features into the same scale. Following this, three (3) techniques are executed to determine the optimal number of principal components (PCs) for the dataset which are as follows:

1. The Cattell-Scree test looks at the plotted eigenvalues. This plot typically shows a sharp bend, or "elbow". PCs with eigenvalues above this sharp bend are kept.
2. Second technique is to take the PCs with a cumulative variance higher than 70%.
3. The Kaiser-Harris criterion recommends taking all PCs with an eigenvalue greater than 1. PCs with an eigenvalue less than 1 explain less information than a single variable in the data.

After conducting all techniques, the optimal number of PCs was determined to be two (2) and the scaled x values were transformed into two (2) PCs, i.e., two (2) dimensions. Following this, a correlation circle was visualised for the two (2) PCs to observe how the predictor variables retained correlation.

As data transformation is complete, the experiment then proceeds to the stage of data splitting or otherwise known as train-test splitting. The data is partitioned into training sets and testing sets of varying ratios, i.e. – 60:40, 70:30, and 80:20 respectively. Training sets are utilised for model training while test sets are used to conduct the actual predictions and their evaluations. The data is stratified by the target variable to preserve equal representation of each target classes. Furthermore, five (5) iterations with distinct random seed values are created for each splitting ratio. This means that the data splitting process is being conducted for a total of fifteen (15) times; five (5) iterations for each of the three (3) varying ratios.

The next step is to conduct the actual training of the models and generate predictions through the Random Forest and Artificial Neural Network algorithms. Starting with Random Forest – it is configured to have a different number of estimators for each of the three (3) varying Random Forest models. This number of estimators essentially refers to the number of decision trees that a particular Random Forest model is to have. The greater the number, the more elaborate the model is.

The Random Forest models are configured as illustrated in the table below:

Model	**Train:Test Split**	**Number of Estimators**
1	60:40	100
2	70:30	550
3	80:20	1000

Subsequently, all fifteen (15) training sets are processed into the Random Forest models to train. Once that process is complete, predictions are generated using the test data splits.

Advancing to Artificial Neural Networks (ANN), an additional data transformation is required. This time, the predictor variables are transformed by a minmax scaler wherein the values are scaled between 0 and 1. Next, like Random Forest, the ANN models need to be configured accordingly. The main configurational change from its default settings was the 'solver' parameter and it was changed from its default value of *'adam'* to *'lbfgs'* as according to ===, the latter performs better on smaller datasets like its being used for this study. Moreover, another configuration was the hidden layer sizes for each of the three (3) varying ANN models. The values for this parameter represent the number of neurons in each hidden layer therefore, the larger the hidden layer size, the more elaborate the model is.

The ANN models are configured as illustrated in the table below:

Model	**Train:Test Split**	**Hidden Layer Size**
4	60:40	100, 100, 100
5	70:30	200, 200, 200
6	80:20	300, 300, 300

Then, like the Random Forest process, fifteen (15) training sets are processed into the ANN models to train. Once that process is complete, predictions are generated using the test data splits.

The final stage of the research pipeline is to analyse results and evaluate the models. The evaluation scores were generated through two (2) sklearn.metric functions – classification_report() and confusion_matrix(), whereby the *y_test* and *predictions* variables serve as parameter inputs for both functions.

Discussion of Results

Figures 1, 2, 3, 4, display the metric scores of the best two performing models and their specific respective data split ratio and model configurations. Model 2 (Random Forest) had a mean accuracy score of 95.2% while model 4 (ANN) had a mean accuracy score of 93.8%.

Random Forest									
Model	**Iteration**	**Type**	**Train:Test**	**n_estimators**	**Accuracy**	**Precision**	**Recall**	**FNR**	**F1-Score**
2	1	RF	70:30	550	0.968253968	0.96969697	0.968253968	0.0952381	0.967829457
2	2	RF	70:30	550	0.952380952	0.954545455	0.952380952	0.1428571	0.952325581
2	3	RF	70:30	550	0.936507937	0.937001595	0.936507937	0.1904762	0.935658915
2	4	RF	70:30	550	0.936507937	0.940648723	0.936507937	0.1904762	0.937157287
2	5	RF	70:30	550	0.968253968	0.96969697	0.968253968	0.0952381	0.967829457
				Model mean:	0.952380952	0.954317942	0.952380952	0.1428571	0.95216014

Figure 1: Random Forest - Model 2 Evaluation Results

As observed in fig. 2, the False Negative Rate of this particular model had varying results. Class0 was incorrectly classified in all five (5) iterations, Class1 was misclassified once in iteration 4, and Class 2 was misclassified three (3) times.

Confusion Matrix				
	Model 2			FNR
	0	**1**	**2**	
0	19	1	1	0.095238095
1	0	21	0	0
2	0	0	21	0
0	20	1	0	0.047619048
1	0	21	0	0
2	2	0	19	0.095238095
0	18	1	2	0.142857143
1	0	21	0	0
2	1	0	20	0.047619048
0	20	0	1	0.047619048
1	2	19	0	0.095238095
2	1	0	20	0.047619048
0	19	1	1	0.095238095
1	0	21	0	0
2	0	0	21	0

Figure 2: Random Forest - Model 2 Confusion Matrix

Artificial Neural Networks									
Model	**Iteration**	**Type**	**Train:Test**	**Layer size**	**Accuracy**	**Precision**	**Recall**	**FNR**	**F1-Score**
4	1	ANN	60:40	100, 100, 100	0.9285714	0.9411765	0.9285714	0.2142857	0.9297552
4	2	ANN	60:40	100, 100, 100	0.9285714	0.9326923	0.9285714	0.2142857	0.9283951
4	3	ANN	60:40	100, 100, 100	0.9642857	0.9677419	0.9642857	0.1071429	0.9646446
4	4	ANN	60:40	100, 100, 100	0.8928571	0.9119048	0.8928571	0.3214286	0.8928571
4	5	ANN	60:40	100, 100, 100	0.9761905	0.9770115	0.9761905	0.0714286	0.9759584
				Model mean:	0.9380952	0.9461054	0.9380952	0.1857143	0.9383221

Figure 3: ANN - Model 4 Evaluation Results

Meanwhile, as observed in fig. 4, ANN models had a higher False Negative Rate across all models and all iterations. Class0 was incorrectly classified in three (3) iterations, Class1 was misclassified in three (3) iterations as well, and Class 2 was misclassified four (4) times in this particular model.

Confusion Matrix				
	Model 4			FNR
	0	**1**	**2**	
0	28	0	0	0
1	4	24	0	0.142857
2	2	0	26	0.071429
0	24	4	0	0.142857
1	0	28	0	0
2	2	0	26	0.071429
0	28	0	0	0
1	2	26	0	0.071429
2	1	0	27	0.035714
0	27	0	1	0.035714
1	7	21	0	0.25
2	1	0	27	0.035714
0	26	1	1	0.071429
1	0	28	0	0
2	0	0	28	0

Figure 4: ANN – Model 4 Confusion Matrix

While better results are always sought for, these misclassifications are justified and somewhat expected since for many of the predictor variables, all three (3) classes lie very close to each other in terms of distribution density, especially class 2 (class 1 in confusion matrix). (Charytanowicz *et al.,* 2010)

Conclusion

In conclusion, by adopting this research pipeline (ref to fig. 5) for the methodology of this study, the research problem was addressed by producing satisfactory results in both algorithms. With a mean accuracy of 95.2% for Random Forest, and 93.8% for Artificial Neural

Networks, these particular algorithms can be expected to deliver favourable results in seed classification. A few notable limitations of this study are:

1) The small size of the dataset.
2) The overlapping distribution of the seeds' predictor variables.
3) The processing power of a domestic machine.

For recommendations, it is advised to address the limitations mentioned above if possible, and to attempt different machine learning algorithms such as Support Vector Machines (SVMs) as that might achieve better results.

Appendix

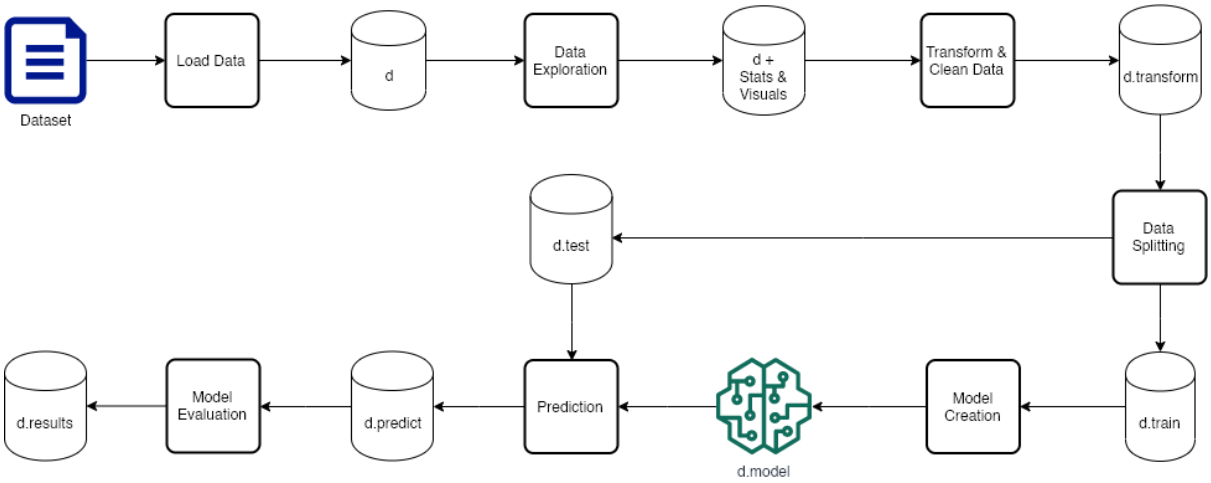

Figure 5: Research Pipeline

References

Charytanowicz, M., Niewczas, J., Kulczycki, P., Kowalski, P.A., Łukasik, S. and Żak, S., 2010. Complete gradient clustering algorithm for features analysis of x-ray images. In Information technologies in biomedicine (pp. 15-24). Springer, Berlin, Heidelberg.