Glass Identification Given Oxide Content: A Comparative Study Of Random Forests And Decision Trees

Supplementary Material

Ayliah Fani

City, University of London Department of Computer Science

Glossary

Accuracy	The fraction of predictions that a classification model got right. In multi-class classification, accuracy is defined as follows: $$accuracy = \frac{correct\ predictions}{total\ number\ of\ examples}$$
Class	One set of enumerated target values for a label.
Classification model	A type of machine learning model for distinguishing among two or more discrete classes.
Confusion matrix	An NxN table that summarizes how successful a classification model's predictions were; that is, the correlation between the label and the model's classification. One axis of a confusion matrix is the label that the model predicted, and the other axis is the actual label. N represents the number of classes.
Continuous feature	A floating-point feature with an infinite range of possible values.
Cross-validation	A mechanism for estimating how well a model will generalize to new data by testing the model against one or more non-overlapping data subsets withheld from the training set.
Dataset	A collection of examples
Decision tree	A model represented as a sequence of branching statements.
Discrete feature	A feature with a finite set of possible values.
Ensemble	A merger of the predictions of multiple models.
Example	Row of a dataset, containing one or more features/predictors and a response/class
Feature	An input variable used in making predictions
Holdout data	Examples intentionally not used ("held out") during training. The validation dataset and test dataset are examples of holdout data. Holdout data helps evaluate your model's ability to generalize to data other than the data it was trained on. The loss on the holdout set provides a better estimate of the loss on an unseen dataset than does the loss on the training set.
Hyperparameter	The "knobs" that you tweak during successive runs of training a model.

Label	In supervised learning, the "answer" or "result" portion of an example. Each example in a labelled dataset consists of one or more features and a label.
loss	A measure of how far a model's predictions are from its label. Or, to phrase it more pessimistically, a measure of how bad the model is. To determine this value, a model must define a loss function.
Loss curve	A graph of loss as a function of training iterations. The loss curve can help you determine when your model is converging, overfitting, or underfitting.
Machine Learning	A program or system that builds (trains) a predictive model from input data. The system uses the learned model to make useful predictions from new (never-before-seen) data drawn from the same distribution as the one used to train the model. Machine learning also refers to the field of study concerned with these programs or systems.
Model	The representation of what a machine learning system has learned from the training data.
Multi-class Classification	Classification problems that distinguish among more than two classes.
Overfitting	Creating a model that matches the training data so closely that the model fails to make correct predictions on new data.
Precision	A metric for classification models. Precision identifies the frequency with which a model was correct when predicting the positive class. That is: $Precision = \frac{true\ positives}{true\ positives + false\ positives}$
Prediction	A model's output when provided with an input example
Random Forest	An ensemble approach to finding the decision tree that best fits the training data by creating many decision trees and then determining the "average" one. The "random" part of the term refers to building each of the decision trees from a random selection of features; the "forest" refers to the set of decision trees.
Supervised Machine Learning	Training a model from input data and its corresponding labels.
Test Set	The subset of the dataset that you use to test your model after the model has gone through initial vetting by the validation set.

Training	The process of determining the ideal parameters comprising a model
Training set	The subset of the dataset used to train a model
Validation	A process used, as part of training, to evaluate the quality of a machine learning model using the validation set. Because the validation set is disjoint from the training set, validation helps ensure that the model's performance generalizes beyond the training set.
Validation set	A subset of the dataset—disjoint from the training set—used in validation.

Definitions taken from Google Developer's Machine Learning Glossary [1]

Intermediate Results

Intermediate results are discussed within the '*MLCoursework.mlx*' file, as well as within this document.

Random Forests

- An initial model was made and performance was assessed through a confusion matrix (fig.1) and classification error calculated at 0.26.
- The model was cross-validated, and out of bag error was calculated. Cross validation loss, out of bag error, and the original model's loss were plotted against the number of trees in the forest (fig.2)
- The model was optimised from values obtained in fig.2 to create the final model and predictions. This final model was evaluated again, through a confusion matrix (fig.3) and classification error calculated at 0.14.

Figure 1: Confusion Matrix of First RF Model

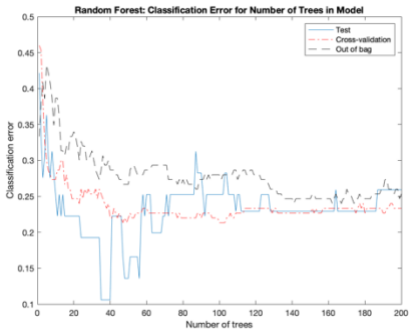

Figure 2: Classification Associated with Number of Trees

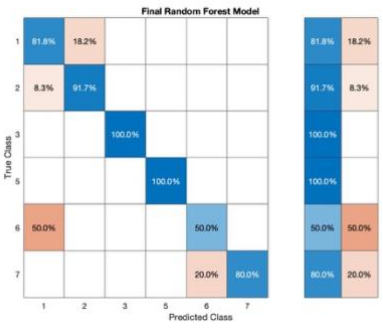

Figure 1: Confusion Matrix of Final RF Model

Decision Trees

- An initial model was trained and performance was assessed through a confusion matrix (fig.4), classification error calculated at 0.37, and cross validation
- The error for each subtree/pruning level for the training and validation set was then assessed and plotted (fig. 5). The loss and confusion matrix was determined for a tree pruned to level 1 (fig. 6).

- The tree was then optimised by selecting the appropriate tree depth. This was done by determining the cross validated error associated with minimum leaf size (fig.7). These hyperparameters were then used to create the final model.
- The final model was made, and again, loss and confusion matrices were used as a performance measures (fig.8). Lastly, the tree was pruned one level to create the final model (fig.9).

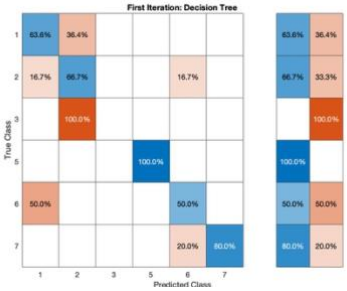

Figure 4: Confusion Matrix of Initial Decision Tree

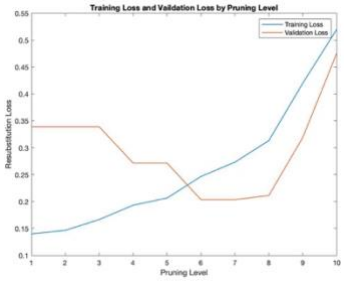

Figure 5: Training and Validation Loss by Pruning Level

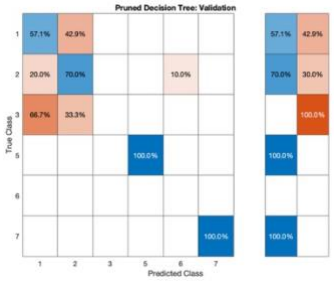

Figure 6: Pruned Decision Tree Confusion Matrix on Validation Set

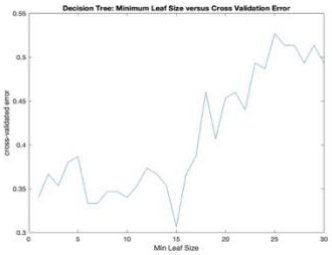

Figure 7: Cross-validated Error Associated with Minimum Leaf Size

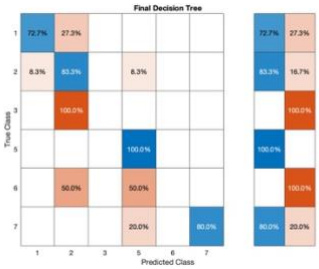

Figure 8: Confusion Matrix of Unpruned Tree

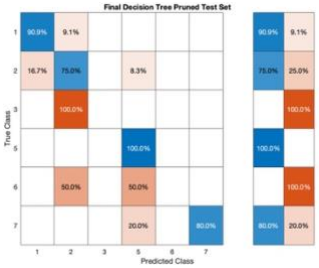

Figure 9: Confusion Matrix of Final Pruned Tree

Implementation Details

Random forests and decision trees were selected for this study. The data set contains continuous features and multiclass responses. From these properties it becomes apparent that this is a classification problem and that regression models do not apply. Additionally, since the dataset is multi-class, classification models such as Logistic regression cannot be applied since they support binary classification problems. Additionally, Naïve Bayes could not be implemented due to some classes having zero variance.

Implementation was done using a single live script for ease of communication and understanding of the process steps.

For random forests an initial model was trained. From that model the number of trees present (number of learning cycles) was chosen as a hyperparameter to be optimised. The loss and error of the model at varying learning cycles was determined and the number of trees yielding the lowest associated error was chosen. The model was cross validated via k-fold cross validation. Finally, the final classifier was trained with the optimised hyperparameters and later implemented.

The same implementation was followed with the decision tree model, although the choice of hyperparameters for investigation and later incorporation into the final classifier was minimum leaf size and pruning level.

Bibliography

[1] ‘Machine Learning Glossary’, *Google Developers*. https://developers.google.com/machine-learning/glossary (accessed Dec. 08, 2020).