Assessment 3: Project

Dominic Shore   
Canberra, Australia  
DominicShore@gmail.com

# Introduction

The task of assessing whether an abalone has had the opportunity to reach sexual maturity and reproduce can be accomplished by “cutting the shell through the cone, staining it, and counting the number of rings through a microscope” [1]. This can be a time consuming and tedious task which must necessarily be performed after the fish has perished. If a model could be developed to assess the age of the abalone by inputting size and weight measurement prior to despatching the fish, it could help regulatory authorities to better calibrate minimum size requirements in order to maximise the value of sustainably harvested abalone subject to the constraint of having given the fish a sufficiently long period to reproduce.

This paper presents a sequence of models have been developed to predict the age range of abalone from physical measurements which can be taken prior to despatching the fish. All models were built with neural networks using Keras [2] and sklearn [3] libraries in python.

# methodology

## Data source, ingestion and transformation

The abalone data was sourced from the University of California Irvine online repository of datasets [4]. In order to ensure that the results were not unduly influenced by the presences of outliers in the data each variable was examined. Using a combination of summary statistics and data visualisation the variable ‘Height’ was found to have two obvious outliers. Those outliers were then removed from the data and subsequent calculations.

Chart

Description automatically generated

1. Outliers in the ‘Height’ variable are shown in the Figure above

The variable of ‘rings’ – denoting the age of the abalone – was converted into a categorical variable as follows by assigning each observation to one of four categories using the following approach:

* Class 1: 0 - 7 years
* Class 2: 8- 10 years
* Class 3: 11 - 15 years
* Class 4: Greater than 15 years

This new variable was named ‘ring\_class’ and will serve as the target variable for the predictions from the models in this paper. To help the models deal with a categorical target variable more easily it was converted using the OneHotEncoder method in sklearn. New variables were created for each category. The ring class variable one-hot encoding can be represented graphically as follows:

A picture containing table

Description automatically generated

1. The transformation of the ring class varaible with one-hot encoding

Similarly, the variable ‘sex’ was transformed using sklearn’s OrdinalEncoder to alter the representation of the flags M (for Male), F (Female), and I (Infant) to 0, 1, and 2 respectively.

Finally, the remaining numerical variables (length, diameter, height, whole weight, shucked weight, viscera weight, & shell weight) were normalised so as to range between the values of 0 and 1. This was achieved with the use of sklearn’s MinMaxScaler.

These transformations were each fit to the data using the Pipeline feature in sklearn. The Pipeline function can be useful as there is often a fixed sequence of steps in processing the data, for example feature selection, normalization and classification [5]. The code that was drawn on to inform the development of the pipeline in use was found in the text for this course [6].

## Data exploration

The distribution of ‘ring\_class’ was found to be contain a large number of observations in Class 2 (8-10 years). This may be because of the previous minimum size restrictions which have historically been placed on the harvest [7] of abalone.

Chart, bar chart

Description automatically generated

1. Distribution of abalone ring count classes

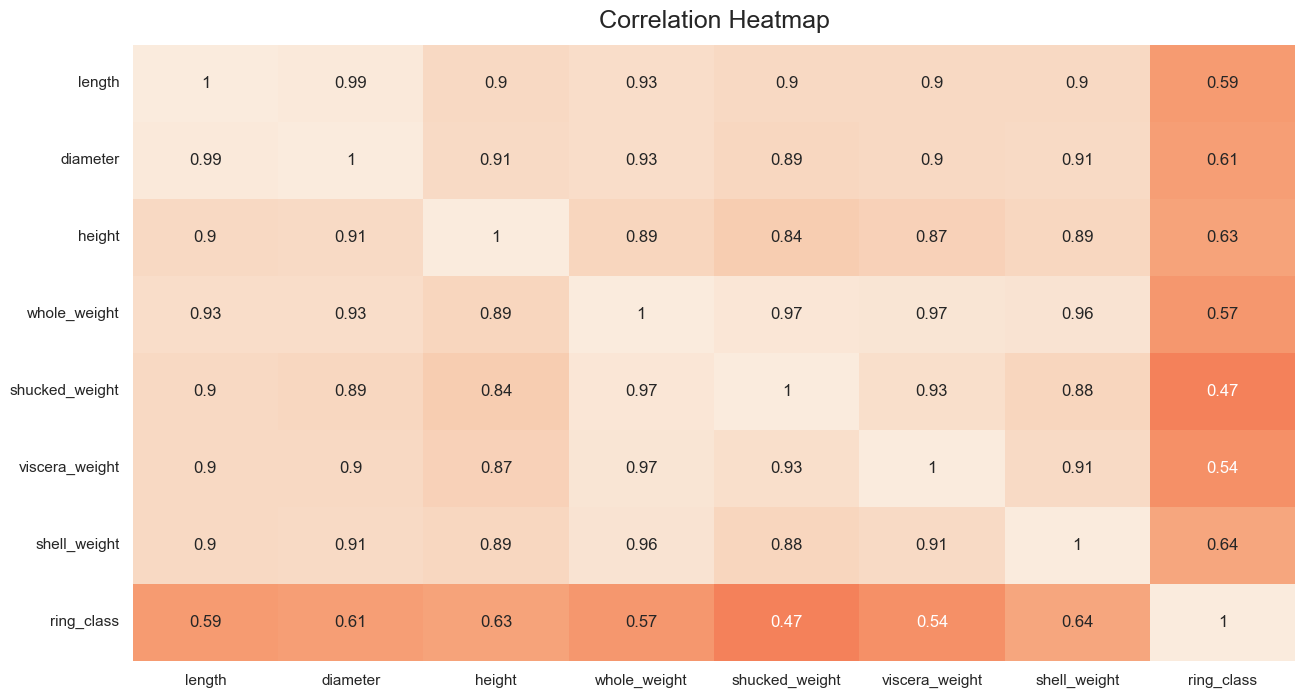
The distribution of each of the variables in the abalone dataset was then examined graphically using histograms:

Chart, bar chart, histogram

Description automatically generated

1. Histogram of each variable included in the abalone dataset from UCI

To understand the relationships between the variables inmore depth a correlation heatmap was developed with the aid of the seaborn library.



1. Correlation heatmap of the abalone dataset from UCI

The correlation heatmap of the abalone dataset showed that the shell height and shell weight were the most highly correlated with the ring class variable.

Ultimately, the dataset was broken into the Training set (which comprised 60% of the data) and the Test set (40% of the data).

## Model contruction

Four related models were constructed to assess the impact of changes in i) hidden neurons, ii) learning rates, iii) hidden layers, and iv) optimisers on the accuracy of the model in predicting the correct ring class.

Each model had an 8-node input layer to accommodate ingestion of the 8 features related to the size, weight and sex of abalone in the dataset.

The initial round of experiments involved changes to the number of neurons in a single hidden layer. These changes were made in an iterative fashion to investigate the effect of differences in model capacity on accuracy. The testing included a hidden layer of size 5, 10, 15, and 20 with each being subject to a *relu* activation function before being passed to the output layer. In all cases, the output layer consisted of four neurons with *softmax* activation which corresponded to the ring class categories of the target variable.

The second round of experiments keeps the optimal number of hidden neurons observed during the initial round constant and makes changes to the learning rate hyperparameter to further optimise the accuracy of the model. The learning rates experimented with in this round were 1e-1, 1e-2, 1e-3, 1e-4, and finally 1e-5.

The third round of experiments investigated the impact of adding a second hidden layer to the network architecture while maintaining the optimal number of hidden layer neurons (from the initial experiment) and the optimal learning rate (from the previous experiment).

Finally, the optimiser was changed in light of previously optimal architecture and hyperparameter settings. In the first three (i – iii) experiments on model accuracy, Stochastic Gradient Descent (SGD) was employed to optimise the weight and bias terms in the neural network. The SGD method entails one random instance of the data at a time which is passed forward through the network of neurons. Next the error resulting from that forward pass is calculated by comparing the actual and desired output from the network. Then, by using the chain rule to calculate the contribution of each output to the total error, the algorithm performs a reverse pass to modify each parameter by moving it by a small fraction in the opposite direction of the gradient calculated using that rule [8].

In this final step the Adaptive Moment Estimator (Adam) was applied to the model. “[Adam] … is perhaps best seen as a variant on the combination of RMSProp and momentum with a few important distinctions. First, in Adam, momentum is incorporated directly as an estimate of the ﬁrst-order moment (with exponential weighting) of the gradient. The most straightforward way to add momentum to RMSProp is to apply momentum to the rescaled gradients. The use of momentum in combination with rescaling does not have a clear theoretical motivation. Second, Adam includes bias corrections to the estimates of both the ﬁrst-order moments (the momentum term) and the (uncentered) second-order moments to account for their initialization at the origin” [9].

In the development of each model 10 experimental runs with different training datasets were performed. The mean and standard deviation of the accuracy metrics where calculated and noted. The hyperparameters from the most accurate model were employed in subsequent models as the outlined above were implemented.

# results

## Hidden neurons

In the first experiment various numbers of neurons were tested in the hidden layer. Each time the experiment was run, a new random sample were taken from in order to construct the Train and Test data sets. This process was repeated 10 times for the different number of neurons. To ensure that the results were comparable each experiment used a learning rate of 0.01.

After performing this process for 5, 10, 15, and 20 neurons in the hidden layer the following results were observed:

Table

Description automatically generated

1. Mean accuracy of 8-20-4 neuron network was 59.7%

The whole testing procedure was run numerous times and the highest accuracy score frequently switched between the 10 and 20 neuron hidden layer networks. However, the 20 layer network did tend to outrank the 10 layer network over repeated tests. The 95% confidence interval for the mean is 0.5801 - 0.6145.

A close-up of a necklace

Description automatically generated with low confidence

1. Representation of the final single layer net with 20 hidden nodes [10]

The 8-20-4 structure depicted in Fig. 7 was ultimately adopted as the basis for the continued experimentation in later questions.

## learning rates

Using the model architecture outlined above the effect of changes in learning rates was investigated. The range of learning rates assessed began at 1e-1 and decremented in 10x steps to 1e-5 (inclusive). Taking the same approach to running the experiment 10 times for each variable as discussed in the previous section and in the Methodology the following maximum mean accuracy was observed when using a learning rate of 1e-2:

Table

Description automatically generated

1. Mean of the accuracy score was greatest at learning rate of 1e-2

The 95% confidence interval for the mean is 0.5750 - 0.6199.

## Layers

Anther hidden layer was added to the network to assess the extent of any resulting change to the level of accuracy observed. Again, 10 tests were run on this new architecture in line with the methodology outlined above.

Table

Description automatically generated

1. An additional hidden layer increases the mean level marginally

The 95% confidence interval for the mean is 0.5844 - 0.6234.

## Optimisers

Until this point the optimiser employed has been the SGD, as discussed earlier in the Methodology section earlier. At this point of the experiment the optimiser was changed to the Adaptive Moment Estimation (Adam) which was, again, accompanied by the 10-fold repetition of the process. The following results were observed of the mean and standard deviation of the accuracy score:

Table

Description automatically generated

1. Mean of the accuracy score was greatest at learning rate of 1e-2

The 95% confidence interval for the mean is 0.6225 - 0.6541.

# Conclusion

The development of this model has involved loading and transforming data, preprocessing of features, data visualisation, and experimentation with model hyperparameters and architectures. Ultimately, the addition of greater model capacity (20 neurons in two hidden layers) and the use of default learning rate parameters coupled with the Adam optimiser yielded the highest mean accuracy metric.

##### References

1. W. J. Nash, “The Population Biology of Abalone (Haliotis species) in Tasmania. I. Blacklip Abalone (H. rubra) from the North Coast and the Islands of Bass Strait” Department of Primary Industry And Fisheries, Technical Report No. 48. 1994
2. F. Chollet, (2015) Keras Volume 10. [keras.io](https://keras.io/)
3. Pedregosa *et al.*, (2011) [Scikit-learn: Machine Learning in Python](http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html), JMLR 12, pp. 2825-2830.
4. Dua, D. and Graff, C. (2019), UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
5. Pedregosa et al., “Pipeline: chaining estimators,” [Pipeline: chaining estimators](https://scikit-learn.org/stable/modules/compose.html#pipeline-chaining-estimators) (accessed on 8 Oct 2022)
6. A. Geron, (2019), Hands on Machine-Learning with Scikit-learn, Keras and TensorFlow pg.69-71, Second Edition. O’Reilly, Canada
7. Fishing Tasmainia, “Abalone Fishing” [https://fishing.tas.gov.au/recreational-fishing/fishing-by-species/abalone/abalone-fishing] (accessed 07/10/2022)
8. D. Rumelhart et al. (1985) “Learning Internal Represenational by Error Propagation,” Defense Tehnical Information Cneter technical report, September.
9. I. Goodfellow, Y. Bengio and A. Courville (2016) “Deep learning,” MIT Press [http://www.deeplearningbook.org] Chapter 8, pg.305
10. A. Le Nail, “Publication-ready NN-architecture schematics,” [https://alexlenail.me/NN-SVG/] (accessed 08/10/2022)