Classification Over Small Datasets with Increasing Cardinality

Brook Queree

Problem Description

This report aims to achieve the following goals:

- Explore the effect of class cardinality on classification accuracy over limited data sets
- Explore the resiliency of a range of machine learning models to the effects of increased cardinality

These experiments were run on the "83_Loeschcke_et_al_2000_Thorax_&_wing_traits_lab pops" dataset. It contains three main classification columns that we will explore:

- Sex, with a cardinality of two, containing distinct values:
  - Male
  - Female
- Species, with a cardinality of two, containing distinct values:
  - D._aldrichi
  - D._buzzatii
- Population, with a cardinality of five, containing distinct values:
  - Binjour
  - Gogango_Creek
  - Grandchester
  - Oxford_Downs
  - Wahruna

Across these classes we have a total of 1731 data points measuring various parts of the fruit fly anatomy. The aim is to produce an accurate classifier that can predict these classes given only the anatomic measurements of a fruit fly data point.

In order to simulate high-cardinality classification these classification columns were also combined into complex appended-class columns with a total cardinality of up to 20 (Species-Population-Sex).

Data Exploration

In order for our classifier to easily differentiate these classes, ideally the distribution of the features will have a significant difference across the classes.

Looking at a number of fly measurement features for the two Sex classes it is clear that the classes have reasonably different distributions across the features.

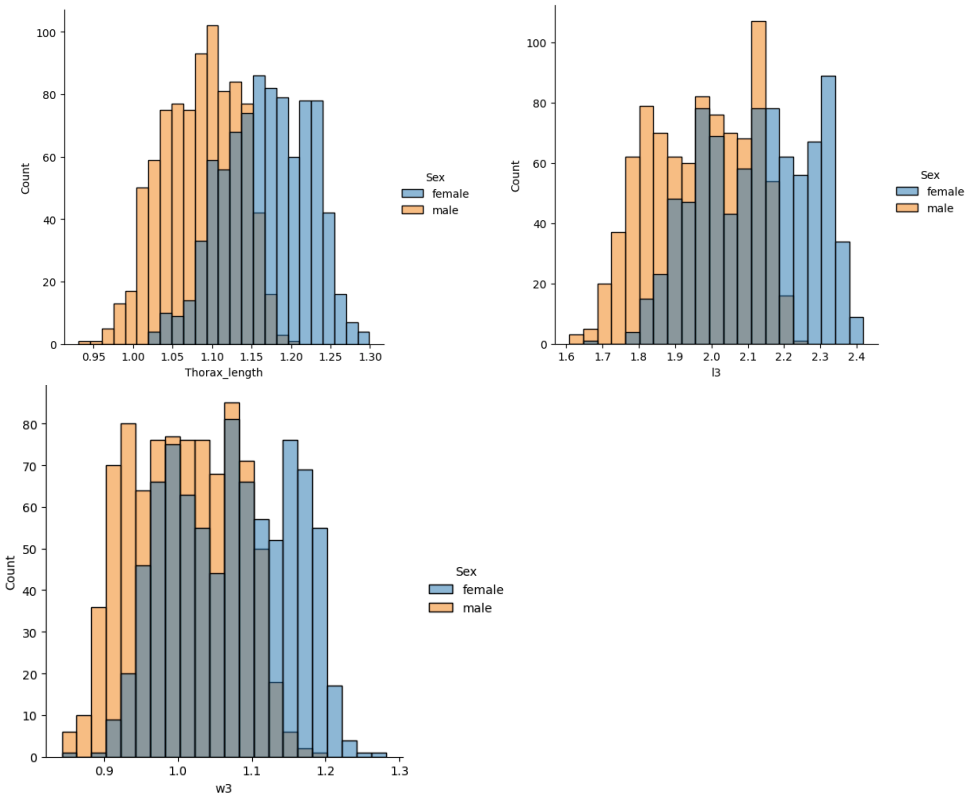
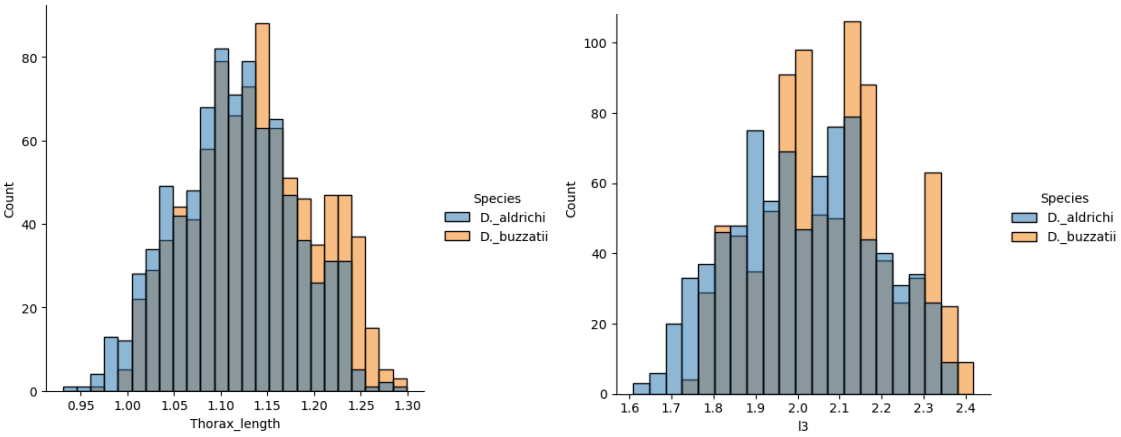


Figure 1: Fruit fly distributions across the Thorax_length, leg (l3), and wing (w3) measurements for the two Sex classes

These partially separate distributions are a good indicator that the fruit fly Sex will be possible to classify with reasonable accuracy.

In comparison, although the Species column also only has a cardinality of 2, the distribution of fly measurements between the two species are much more similar:



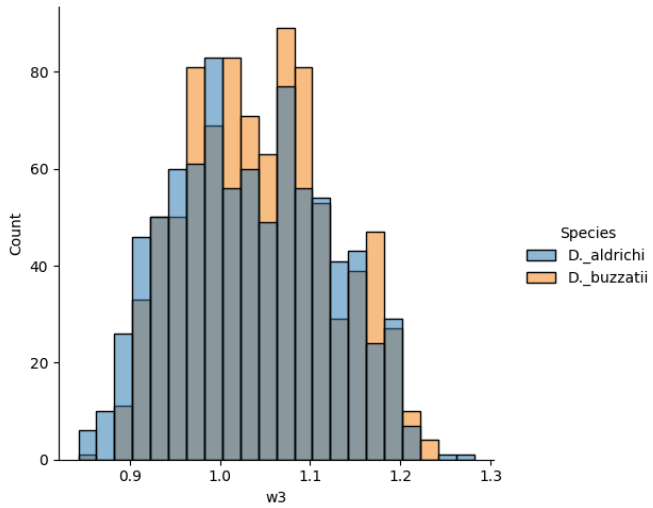


Figure 2: Fruit fly distributions across the Thorax_length, leg (l3), and wing (w3) measurements for the two Species classes

This indicates that we are much less likely to produce an accurate classifer for the Species property due to the similarity of fly anatomy between the two species.

In the third higher-cardinality class column Population we see not only are the distributions near-identical, the lack of data points start to really hamstring the ability to distinguish between classes.

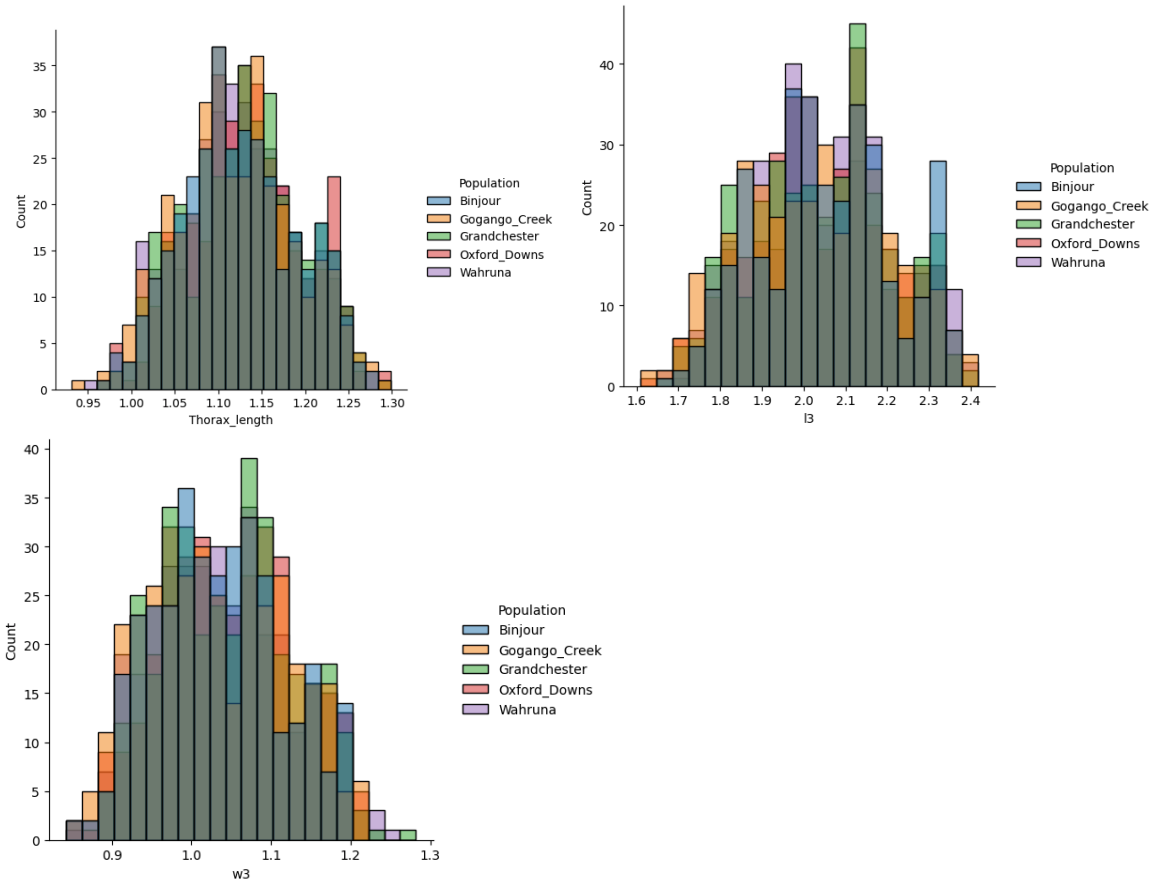


Figure 3: Fruit fly distributions across the Thorax_length, leg (l3), and wing (w3) measurements for the five Population classes

It is possible that there are slight variations between the distributions here, but without more data points it is near-impossible to notice any significant variations between the distributions from the naked eye. This indicates that it is unlikely that we will be able to produce a highly-accurate model of Population from the given fly measurement data.

Data Cleaning

Exploring the data we notice that there is a singular row in the 83_Loeschcke dataset that contains neither a Thorax_length value, nor a wing_loading value. Both simply contain the character ".".

```
erroneous = lab_data[lab_data['Thorax_length'] == '.']
erroneous
```

	Species	Population	Latitude	Longitude	Year_start	Year_end	Temperature	Vial	Replicate	Sex	Thorax_length	l2	l3p	l3d	lpd	l3	w1	w2	w3	wing_loading
253	D._aldrichi	Gogango_Creek	-23.77	150.17	1994	1994	25	6	3	female	.	1.65	0.573	1.415	1.988	1.986	0.898	1.207	1.003	.

Figure 4: Missing Thorax and wing-loading data point

Due to the small nature of the data set and the fact that all other measurement values are available for this row, it makes sense to infer a sensible placeholder for the Thorax_length and wing_loading so as not to throw away data.

For this purpose, the median Thorax_length across the 87 rows containing identical Species, Population, and Sex was used as an estimate of Thorax_length for this specimen. From there, the wing_loading estimate was calculated as per the definition specified in Loeschcke et. al.

$$wing\_loading = \frac{L3}{Thorax\_length}$$

This strategy gives us an estimated Thorax_length of 1.141, and an estimated wing_loading of 1.741 for the partially complete row.

There are an additional two rows that were majority empty/invalid containing zero for seven and nine numeric feature columns respectively:

```
erroneous_2 = lab_data[(lab_data['l3d'] == 0) & (lab_data['l3p'] != 0)]
erroneous_2
```

	Species	Population	Latitude	Longitude	Year_start	Year_end	Temperature	Vial	Replicate	Sex	Thorax_length	l2	l3p	l3d	lpd	l3	w1	w2	w3	wing_loading
61	D._aldrichi	Binjour	-25.52	151.45	1994	1994	25	3	1	female	1.106	0.0	0.6	0.0	0.0	0.0	0.0	1.252	0.0	0.000

[27]:
```
erroneous_3 = lab_data[(lab_data['l3d'] == 0) & (lab_data['l3p'] == 0)]
erroneous_3
```

[27]:

	Species	Population	Latitude	Longitude	Year_start	Year_end	Temperature	Vial	Replicate	Sex	Thorax_length	l2	l3p	l3d	lpd	l3	w1	w2	w3	wing_loading
698	D._aldrichi	Wahruna	-25.2	151.17	1994	1994	20	5	3	female	1.151	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000

Figure 5: Erroneous zero-value data points

We make an assumption here that these records are incorrectly completed. This assumption is made on the predicate that it does not seem physically possible for a fly wing or leg to have zero length in some dimensions but nonzero length in others. A missing leg or wing could explain zero values in all dimensions but mixed-values seem physically impossible.

To resolve this, similarly to above, the zero-value columns were estimated to be the median of the associated column across all rows of the same (Species, Population, Sex). The non-zero measurements were assumed to be measured correctly. Rows 61 and 698 were hence transformed to fully-complete rows as shown in figure X:

	Species	Population	Latitude	Longitude	Year_start	Year_end	Temperature	Vial	Replicate	Sex	Thorax_length	l2	l3p	l3d	lpd	l3	w1	w2	w3	wing_loading
61	D._aldrichi	Binjour	-25.52	151.45	1994	1994	25	3	1	female	1.106	0.0	0.6	0.0	0.0	0.0	0.0	1.252	0.0	0.0

	Species	Population	Latitude	Longitude	Year_start	Year_end	Temperature	Vial	Replicate	Sex	Thorax_length	l2	l3p	l3d	lpd	l3	w1	w2	w3	wing_loading
61	D._aldrichi	Binjour	-25.52	151.45	1994	1994	25	3	1	female	1.106	1.81025	0.6	1.525375	2.122125	2.121125	0.923875	1.252	1.050625	1.917835

	Species	Population	Latitude	Longitude	Year_start	Year_end	Temperature	Vial	Replicate	Sex	Thorax_length	l2	l3p	l3d	lpd	l3	w1	w2	w3	wing_loading
698	D._aldrichi	Wahruna	-25.2	151.17	1994	1994	20	5	3	female	1.151	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

	Species	Population	Latitude	Longitude	Year_start	Year_end	Temperature	Vial	Replicate	Sex	Thorax_length	l2	l3p	l3d	lpd	l3	w1	w2	w3	wing_loading
698	D._aldrichi	Wahruna	-25.2	151.17	1994	1994	20	5	3	female	1.151	1.7405	0.6	1.4925	2.0855	2.084	0.943	1.2805	1.0645	1.810599

Figure 6: Zero-value data points before and after population with estimated dimensional data

Decision Tree Classification

The lab data was first classified using the simple non-parametric decision tree classifier.

A test set containing 20% of the lab data was put aside to evaluate the model accuracy on new data. Although this represents a significant portion of the small lab data set, it allows us to determine an accurate estimate of the model classification accuracy and generalisation when encountering new data points.

The model was trained as a classifier for Sex using four hyperparameter configurations. One with a max tree depth of 5, another with a max depth of 10, one with a fully fit decision tree, and one restricted with a minimum of 3 data points per leaf of the tree.

Hyperparameter Configuration	**Test Set Classification Accuracy**
Max tree depth 5	0.82
Max tree depth 10	0.79
Fully fit tree	0.77
Minimum of 3 leaf data points	0.78

As the tree with max-depth of 5 performed best over the data provided in the Sex classification problem, the same hyperparameters were used going forward for the remaining classifiers.

Given the overlap in distribution of fly features between the Sexes, combined with the limited training data, obtaining an 82% accurate model on fruit fly Sex classification represents a significant success.

These model hyperparameters were used against the same data set to predict classes of increasing complexity and cardinality.

Class	**Cardinality**	**Test Set Classification Accuracy**	**Random Classifier Accuracy**
Sex	2	0.82	0.5
Species	2	0.65	0.5
Species-Sex	4	0.51	0.25
Population	5	0.23	0.2
Species-Population	10	0.13	0.1
Population-Sex	10	0.18	0.1
Species-Population-Sex	20	0.11	0.05

We can see as the cardinality of the class increases, the accuracy of the model decreases. A significant effect is also seen in the presence of the Sex class in the combined-class fields, as this field has a much higher individual class accuracy than the others due to its more distinct distributions.

Some of the poorly performing high cardinality classification models tend towards a random classifier's accuracy. i.e. they are not significantly more accurate than randomly guessing the class of a data point.

In order to explore the effect of hyperparameters on model accuracy at high cardinalities, the same hyperparameter configurations were evaluated against the Species-Population-Sex classification problem:

Hyperparameter Configuration	**Test Set Classification Accuracy**
Max tree depth 5	0.11
Max tree depth 10	0.13
Fully fit tree	0.14
Minimum of 3 leaf data points	0.15

This seems to indicate that more complex models perform better at higher cardinality with limited data. Due to the limited data available in each class as cardinality increases, it seems unlikely that the complex models reach the overfitting state that

appeared to hold them behind the limited-depth model in the previous hyperparameter exploration on the simple Sex class.

Softmax Classification

A softmax classifier was also explored on the same dataset problems.

The previous decision tree classifier, which looks purely at data point boundaries to calculate error and produce rules, did not need data scaling to work optimally.

A standard logistic regression (the base unit of softmax regression) should also not inherently require data scaling to converge to a likelihood maxima. However we are going to explore regularisation with this classifier, which can be effected by the scale of the features while it tries to regularise the parameters.

As a result of this, a min-max scaler was used to normalise both the training and test data to the range of the training data between 0 and 1. i.e. The maximum value of each feature in the training data was taken as value 1.

Similarly to above, the Sex classification task was used as a benchmark to perform hyperparameter exploration. In this case, the hyperparameter updated (C) was the inverse-regularisation-strength. This parameter controls the L2 regularisation being performed on the model. Where lower values of C represent stronger regularisation, and higher values of C reduce the regularisation effect.

Hyperparameter Configuration	**Inverse Regularisation Strength Hyperparameter**	**Test Set Classification Accuracy**
Non-regularised softmax	N/A	0.86
Strong-regularised softmax	0.01	0.72
Medium-regularised softmax	1	0.85
Weak-regularised softmax	100	0.86

This outcome was surprising, as typically regularisation is expected to reduce E_{new} (the expected error over new unseen data points). We can theorise here then that the relatively small dataset size and feature count means that the softmax classifier does not have enough data to reach a point of overfitting.

Another hyperparameter that could be explored on a larger data set is the maximum training iterations. Due to the small nature of this data set the softmax classifier was already converging on the Sex classification problem within 20 iterations, and restricting this further had the effect of reducing the test set accuracy. As such the effects of this hyperparameter could not be fully explored on this problem.

As a result, the remaining class cardinality problems were explored without regularisation.

Class	Cardinality	Test Set Classification Accuracy	Random Classifier Accuracy
Sex	2	0.86	0.5
Species	2	0.70	0.5
Species-Sex	4	0.65	0.25
Population	5	0.29	0.2
Species-Population	10	0.21	0.1
Population-Sex	10	0.24	0.1
Species-Population-Sex	20	0.16	0.05

We can see that there is a noticeable improvement in classification accuracy over the test set compared to the decision tree classification above.

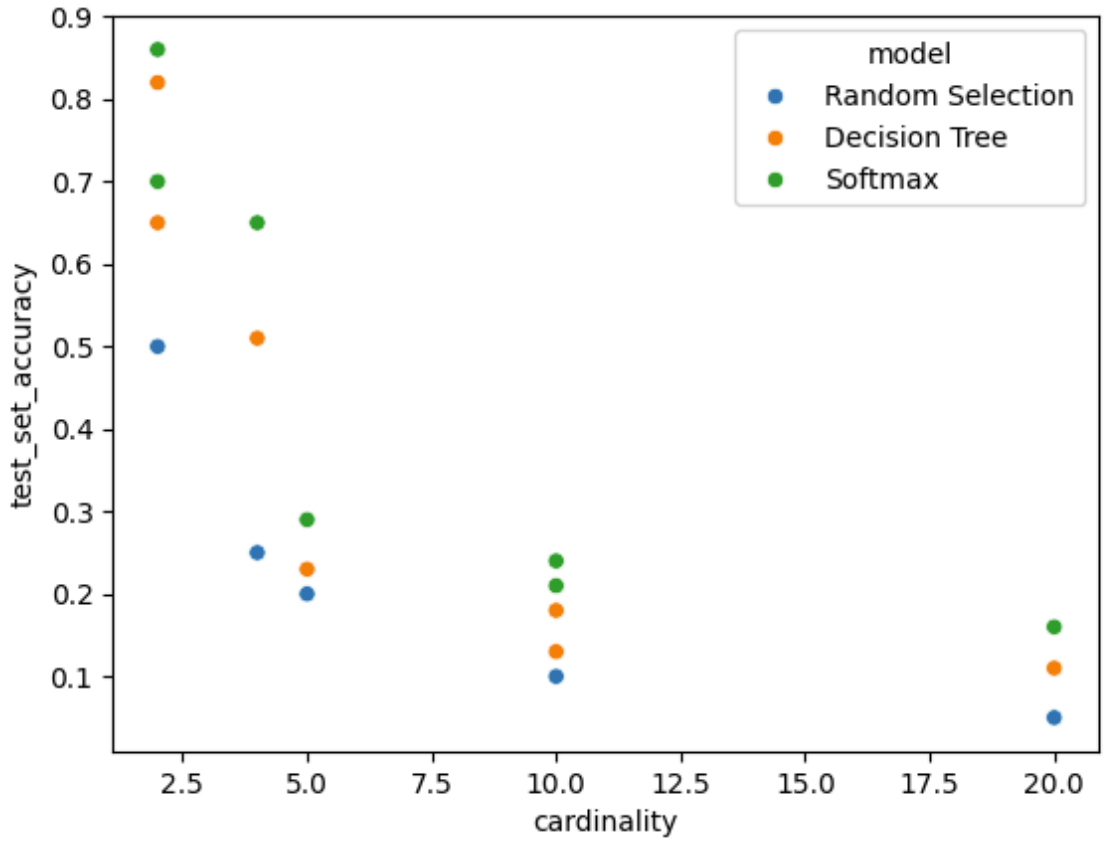


Figure 7: Model performance on differing-cardinality classification problems

As before, we can again see that the cardinality of the class problem appears correlated to the classifier accuracy, as well as the presence of the Sex class in the combined-class features.

The various hyperparameter configurations were again explored against the high cardinality Species-Population-Sex combined class feature.

Unlike the low-cardinality problem, we could also explore the maximum iteration hyperparameter on this high-cardinality classification as the classification problem took many more iterations to converge. Note that the readings taken above were extracted by using the library default of 100 maximum iterations. Here we explore the effect of increasing and decreasing this parameter.

Hyperparameter Configuration	**Maximum Training Iterations Hyperparameter**	**Test Set Classification Accuracy**
Non-regularised softmax	20	0.14
Non-regularised softmax	100 (default)	0.16
Non-regularised softmax	200	0.16
Non-regularised softmax	400 (converged)	0.16

Again, it seems the lack of training data restricts our ability to explore this hyperparameter. However, this gives us confidence that the training iterations hyperparameter had little effect on the previous results.

The same hyperparameter exploration as in the Sex classification problem was again re-explored in the same vein as we performed on the decision tree classifier.

Hyperparameter Configuration	**Inverse Regularisation Strength Hyperparameter**	**Test Set Classification Accuracy**
Non-regularised softmax	N/A	0.16
Strong-regularised softmax	0.01	0.05
Medium-regularised softmax	1	0.13
Weak-regularised softmax	100	0.16

This time the hyperparameter results were consistent across both the low cardinality and high cardinality classification problems. It's interesting to note that both the decision tree classifier and the softmax classification saw improved performance at high cardinality by increasing the model complexity (regularisation having the effect of reducing model complexity).

On observing this increased-complexity benefit at high cardinalities we can theorise that potentially very-complex models such as neural networks may see improved results as we increase the cardinality of classes and difficulty of the classification problem.

Neural Network Classification

Given our exploration above, it's possible that the flexibility of a neural network is better able to handle the complexity of the higher-cardinality classification.

As in the softmax classification, the features were normalised to the zero-to-one range. In neural network training this has the effect of ensuring that the gradient used in adjusting the parameters during training are all of the same scale. This ensures that the gradient descent performed during training won't drastically descend towards specific features due to the scale of their values.

Restricted Linear Units (ReLU) were used as the base neuron activation function for the neural network. This was chosen due to the computational efficiency ReLU offers over the sigmoid activation function. This can allow us to create larger networks, modelling more complex relationships between variables while remaining feasible to train on the single CPU that these tests are being performed on.

As this is a classification problem, the final layer of the neural network we are building will be a dense softmax layer with the number of units equal to the number of classes we are modelling.

The Adam (Adaptive Moment) optimisation algorithm was chosen due to its ability to converge quickly in complex networks. In comparison to Stochastic Gradient Descent, Adam is generally understood to converge faster but be less accurate when generalising. In this scenario where we have limited training data and limited training epochs we are unlikely to overfit to the point of not generalising well, as such we can optimise for CPU efficiency here again as per the ReLU vs. logistic function decision.

There are many hyperparameters involved when training neural networks, and their effects can be unclear over even similar problem sets. Because of this, a range of network architectures were used on each classification problem to explore their performance with respect to cardinality.

As a first guess, a network with two dense ReLU layers containing 1024 units followed by the softmax layer was used to experiment with the number of training epochs to use. If the training is run too long, the network can overfit the training data, whereas training not long enough can have the inverse effect of underfitting.

The test set accuracy on the Sex classification problem was plotted over 5000 epochs to check initial performance (where an epoch is a single pass over the training data).

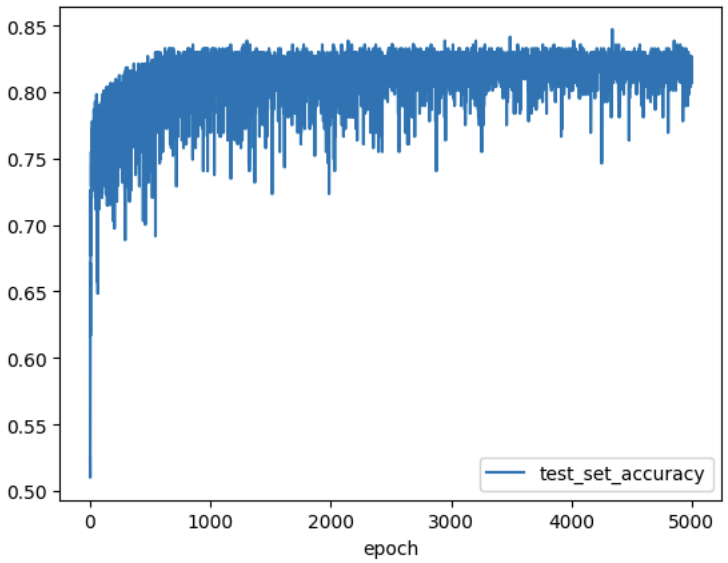


Figure 8: Neural network model accuracy during Sex classification training

We can see that the test set accuracy after convergence is quite noisy, the test set accuracy appears to change drastically during training, even after the model has mostly converged. This in conjunction with the aggressive gradient on the initial training epochs could indicate that we may have set the learning rate too high during training.

Decreasing the learning rate drastically from 1e-3 to 1e-5 we can see that the noise caused by drastic parameter swings during training was significantly reduced, however the model had still not yet converged to the higher peak accuracy as seen in the previous training.

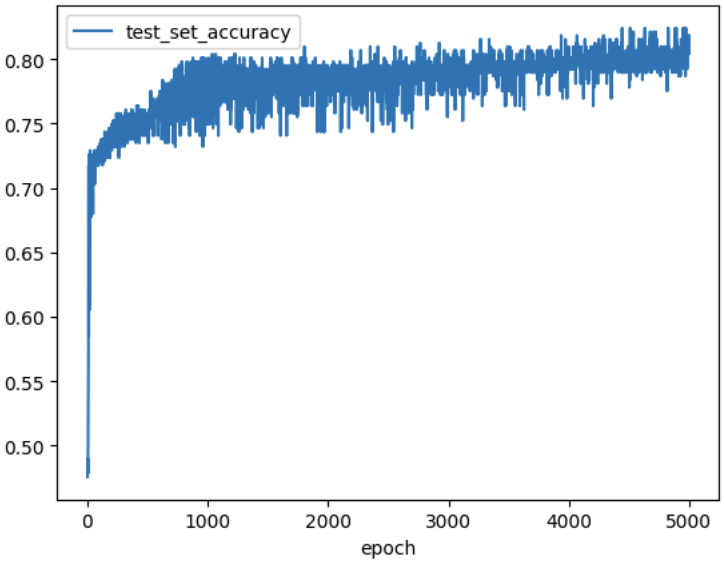


Figure 9: Decreased learning rate neural network model accuracy during training

To balance these two effects a learning rate of 1e-4 was chosen for subsequent training. Using this learning rate, we can see the benefits of the increased training speed while also reducing noise in the parameter swings.

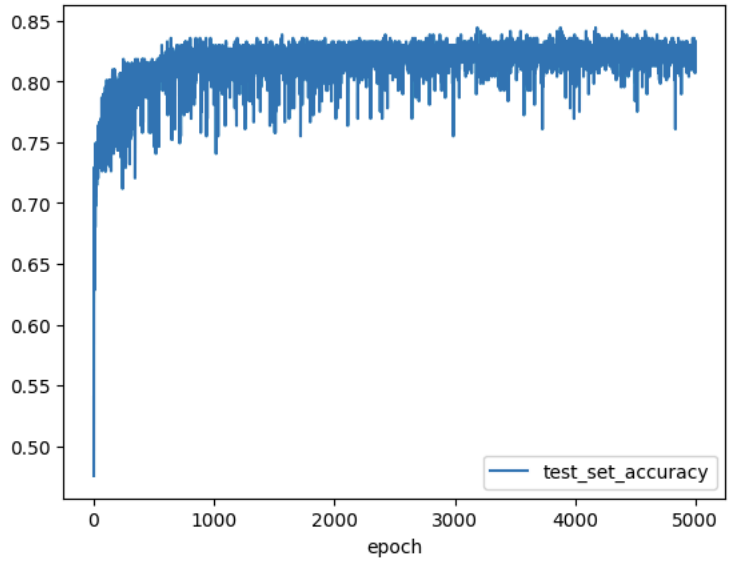


Figure 10: Final learning rate neural network model accuracy during training

We can see now that the model converges within ~1000 epochs, and the accuracy drops between epochs have had a noticeable decrease compared to the initial learning rate.

In this report we are effectively solving many different classification problems. These different problems change significantly enough across cardinalities to potentially warrant different hyperparameters. The same learning rate and model architecture was run against the cardinality 20 Species-Population-Sex classification problem to ensure the model would still converge to an accurate result with this learning rate.

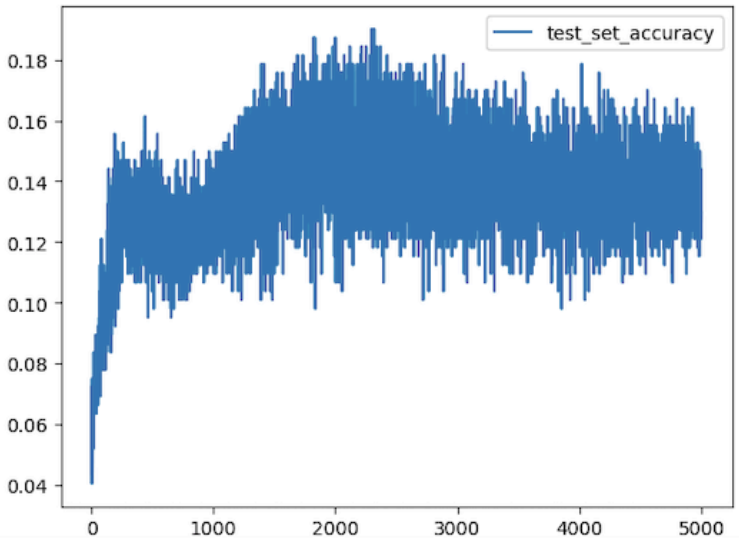


Figure 11: Neural network model accuracy during Species-Population-Sex classification training

Running the training on a 1e-4 learning rate seemed to have the same noise and drastic performance between epochs issues as the 1e-3 learning rate on the previous problem.

To check if this could be resolved using the same strategy a 1e-5 learning rate was tested on the same classification problem.

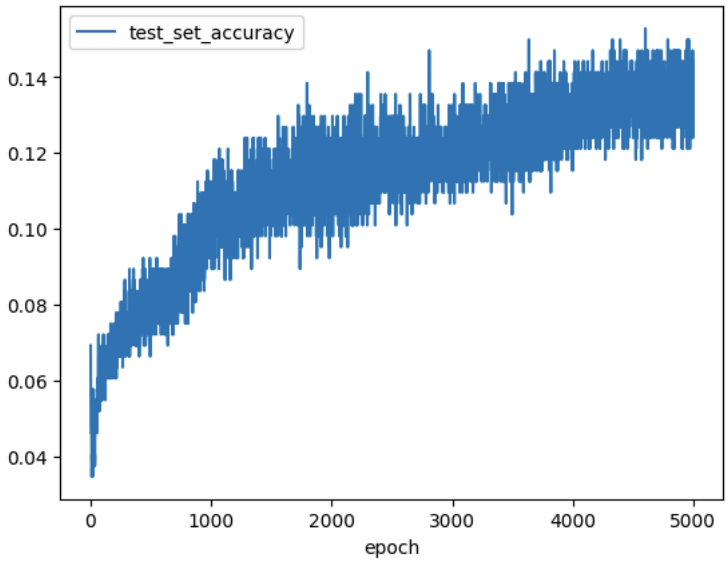


Figure 12: Decreased learning rate neural network model accuracy during Species-Population-Sex classification training

Running for 5,000 epochs we can see that the noise in the test set accuracy was greatly decreased. Although importantly the model appears to still not have converged after this time with the lower learning rate. Considering the higher learning rate still ultimately converged to a similar accuracy in a much shorter time, it does not seem preferable to use the lower learning rate here either.

To that effect, the learning rate was kept at 1e-4 across all classification scenarios for consistency in comparisons. We can consider this reasonable as it has been shown to perform moderately well across the data set features in both low and high cardinality classification scenarios.

To completely explore the effect the number of training epochs has on the model, training was run for 25,0000 epochs across the complex Species-Population-Sex classification problem. This allows the model to trend towards completely fitting the training data. The mean of the last 10 epoch runs was plotted to smooth out the noise in accuracy seen above.

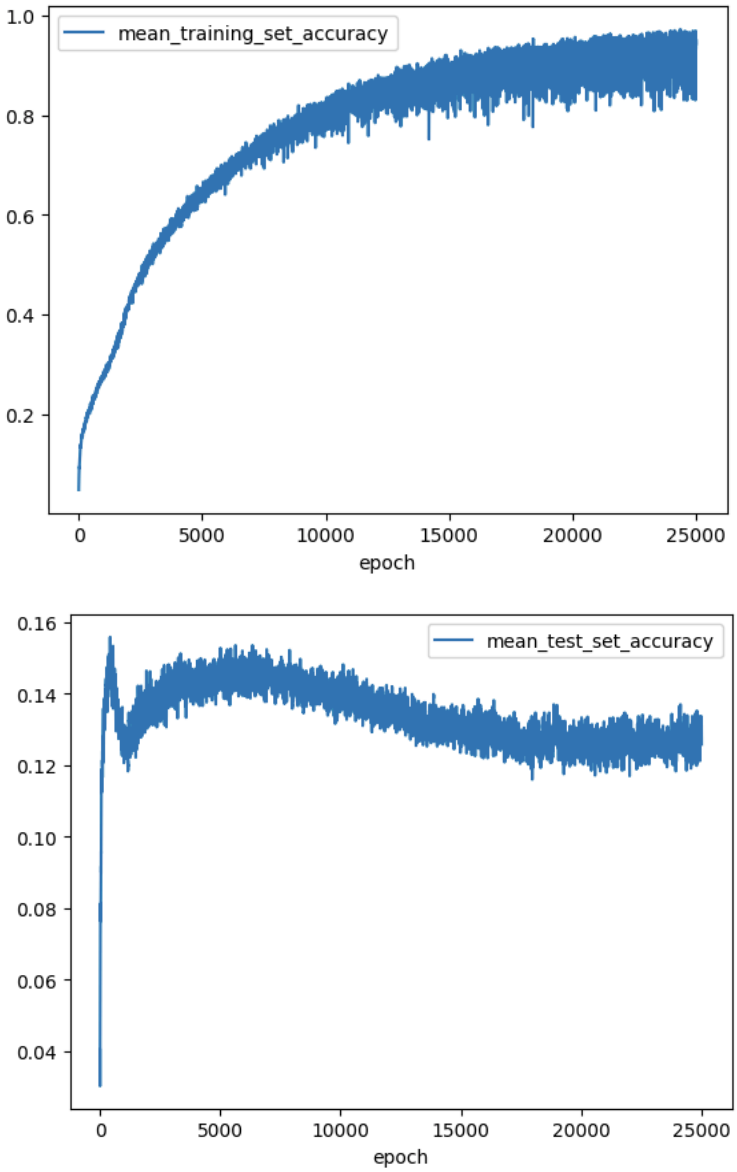


Figure 13: Training and test set accuracy of neural network model during Species-Population-Sex training towards complete overfitting

We can see as before while the model training set accuracy continues to increase up to near 100%, the model accuracy on the test set degrades over time. It appears as if there is an optima reached before 2000 epoch, as was also seen simple Sex classification problem. Using this information, an epoch total of 1200 was chosen as a consistent hyperparameter across the classification problems.

Different network architectures were also experimented with. Because dense neural networks lack easy visibility of their internal connection parameters, it's hard to tell the effects of architecture changes as the classification problem changes. As such, we will run the classification problems on four network architectures, ranging from less to more complex.

Architecture	Layer 1	Layer 2	Output Layer

128 Architecture	128 ReLU units	128 ReLU units	Softmax classifier M units
512 Architecture	256 ReLU units	256 ReLU units	Softmax classifier M units
256 Architecture	512 ReLU units	512 ReLU units	Softmax classifier M units
1024 Architecture	1024 ReLU units	1024 ReLU units	Softmax classifier M units

Where in the table above M represents the cardinality of the class being modelled. Using these model architectures the following test set accuracy results were produced.

Class	**Cardinality**	**Test Set Classification Accuracy**	**Random Classifier Accuracy**
Sex	2	128: 0.83 256: 0.83 512: 0.82 1024: 0.83	0.5
Species	2	128: 0.69 256: 0.68 512: 0.68 1024: 0.67	0.5
Species-Sex	4	128: 0.56 256: 0.55 512: 0.55 1024: 0.53	0.25
Population	5	128: 0.26 256: 0.24 512: 0.19 1024: 0.18	0.2
Species-Population	10	128: 0.16 256: 0.19 512: 0.19 1024: 0.16	0.1
Population-Sex	10	128: 0.23 256: 0.23 512: 0.21 1024: 0.22	0.1
Species-Population-Sex	20	128: 0.15 256: 0.14 512: 0.12 1024: 0.14	0.05

We can see that the model architecture appears to have little effect on the final accuracy result. It is possible that this is because even the 128-unit architecture is highly over-parameterised for this problem Additionally, there is limited total data to

train on, limiting the overall information learnt by the network. The fact that many of the classification accuracy results decrease in accuracy as we increase the unit count in the architecture indicates that this may be the case.

Interestingly, even being extremely generous and selecting the best-performing architecture per classification problem, the neural network model is not able to outperform the softmax model accuracies in any of the classification problems seen previously.

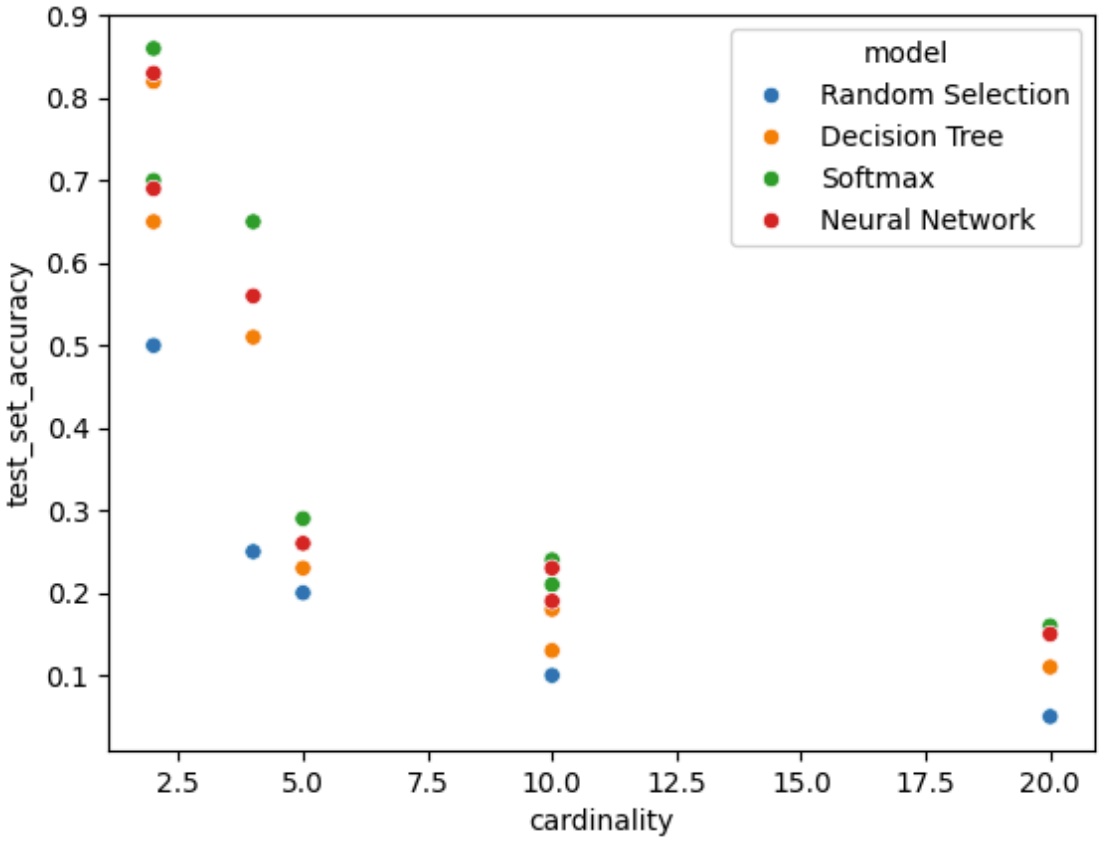


Figure 14: Model performance on differing-cardinality classification problems

We can theorise that this is because although the neural network models offer increased model complexity, the limitations of the data mean that they are not able to learn more complex relationships above what softmax classification is already capable of learning.

Conclusion

As we have explored in the previous model training process, when training on a fixed small number of data points the classification problem becomes more difficult as the cardinality of the class being modelled increases.

In conjunction with the lower overall likelihood of randomly classifying correctly, as the number of classes increases the data point count to train on per class decreases. We

can see this effect in our training results where the test set accuracy both decreases with cardinality and the delta between the model accuracy and random selection decreases.

This was also seen to be affected by the ease of classification of the data distribution itself, with easily separable class distributions (i.e. Sex) more likely to be classified accurately.

As we have explored, a softmax classification algorithm performs very well in these scenarios, even with increased class cardinality. This outcome is likely explained by both the simplicity of the algorithm as well as the data set. The increased complexity of neural networks are not able to learn any new information above what the softmax classification is already learning from the data. Additionally, the decision tree classifier is likely held back by both lack of data points as well as the lack of variance between the distribution across most classes in the features provided.