May 15, 2024

COMP4702 - Machine Learning: The effectiveness of body metrics on the prediction of Sex in Drosophila aldrichi and D. buzzatii

by

Jack Cashman: 47431748

Abstract:

We consider the binary classification problem of predicting the sex of various Drosophilia aldrichi and D. buzzati based on a plethora of quantitative measurements describing physical features of 1704 distinct flies. In order to conduct this analysis, the report will include 5 chapters. (1) Introduction, Data Cleaning and EDA, (2) Training of a logistic regression model, (3) Training of a gradient boosting classifier, (4) Training of a neural network, and (5) A conclusion. In all of these, we investigate relevant hyper-parameter tuning and regularisation techniques.

– 2024

1 Introduction

The *Drosophilia aldrichi* and *D. buzzatii* are species of fruit fly. The dataset presented for analysis consists of a variety of features, both numerical and categorical, of approximately 1700 of these flies that were captured along a latitudinal transect spanning 800 km across Queensland. In the ensuing report, we address the question *'To what extent can the sex of the Drosophilia aldrichi and D. buzzatii be predicted utilising a variety of body measurements'*.

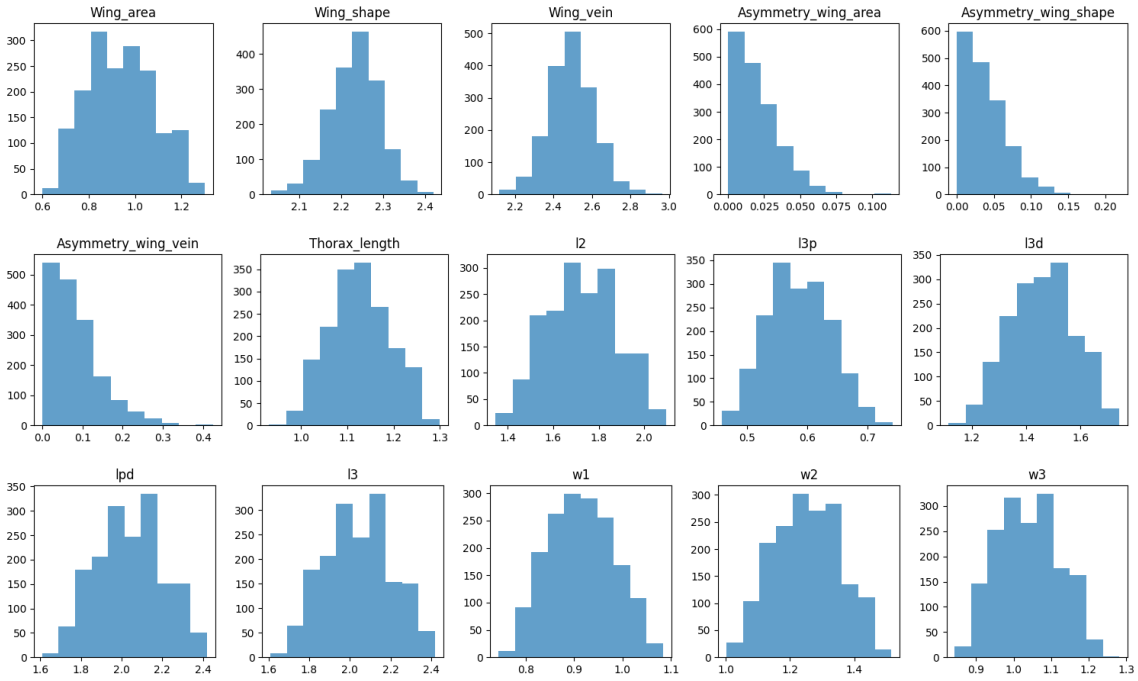
In this report, we take the approach of building binary classification models of increasing computational complexity. Namely, we begin with the development of a simple logistic regression model, then to a gradient boosting classifier, before concluding with a feed forward neural network. Despite the presumably inferior predictive power, the Logistic Regression model is advantageous given the fact that we gain some interpretative insight as to which features have significant impact onto the sex of the fruit fly, and which are less important. Often, this interpretability is thought of as the distinction from *statistical* learning and *machine* learning. For each model $\mathcal{M}$, we provide an estimate of the classification error $E_{\mathcal{M}}$ experienced by the model on new, unseen data, which will be used as a means of distinguishing the effectiveness of each model.

Before the models can be trained, some data pre-processing must be completed. Namely, the data was delivered to us in the form of 3 separate CSV files. The csv file pertaining to the `Wing_asymmetry` had a different number of rows than the other 2 files, so was discarded as we could not ensure that this data could be combined correctly with the other files. The other 2 CSV files were loaded into a dataframe using the `pandas` library. Now, incomplete values were removed, and a mapping female $\mapsto 0$, male $\mapsto 1$ was applied to encode our categorical labels. Furthermore, it was noticed that few entries in the `Thorax_length` and `wing_loading` columns were not consistent with the existing format, so they were dropped from the dataset. Once this pre-processing was completed, we have a dataset with 15 explanatory variables, and 1704 rows will be used in the prediction of sex. In the dataset, we partition the data as follows:

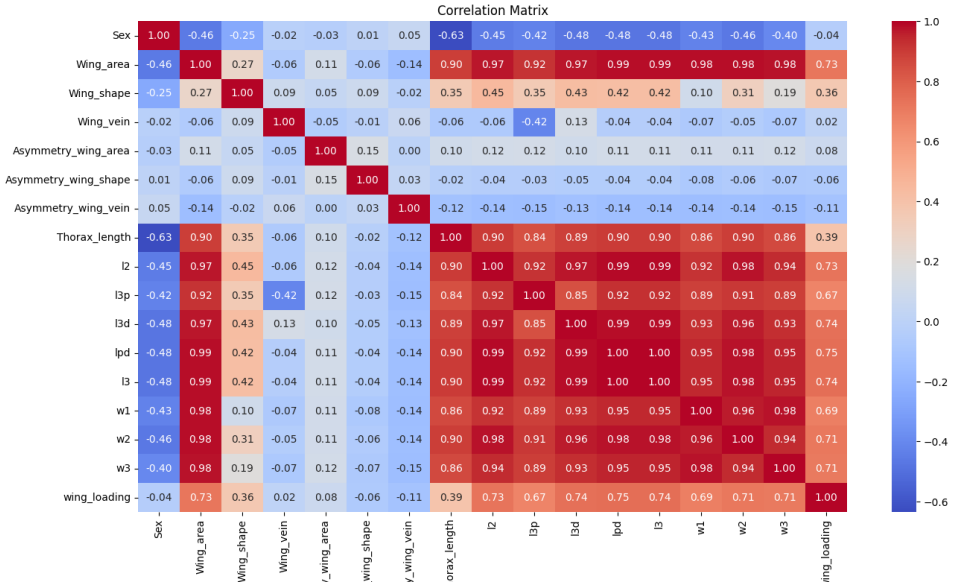
- 70% of the data will be used to train each model. For hyper-parameter selection, we will perform cross-validation on this dataset.

- 15% of the data will be used to produce an unbiased estimation of the generalisation error for each model, so that we can compare models.
- 15% of the data will be used to produce an unbiased estimation of the generalisation error of the 'best' model after comparison between all models.

An initial plot of the data produces the following results:



For the most part, 12 of the data columns appear to be approximately normally distributed. However, the exception to this are the measurements relating to any of the Asymmetry measurements. This is logical, as the notion of a 'negative' asymmetry is nonsensical. None of the models we investigate will make assumptions on the distribution of our explanatory variables, so no further investigation is needed here. However, we naturally may wonder how the variables are correlated with one another. To investigate this, we construct a correlation matrix amongst the data:



Clearly, most of the measurements pertaining to the flies are heavily correlated, due to the nature of the measurements themselves. For example, a fly with a very large left wing would, presumably, also have a very large right wing. Now, this correlation is not problematic, but it is certainly something that will be noted. Indeed, we could perform PCA to produce a set of linearly independent variables, but that would result in the co-efficient of our Logistic regression model to have no physical value, which is not desirable at this point.

2 Logistic Regression Model

We will conduct our logistic regression at the predefined significance level $\alpha = 0.1$. That is, any explanatory variables that have associated a p-value greater than 0.10 will be pruned from the model, and it will be re-trained. To conduct this, we use the `statsmodels.api`, an open source statistical computing library. Using the module outputs to following code:

Logit Regression Results

Dep. Variable:	Sex	No. Observations:	1192
Model:	Logit	Df Residuals:	1176
Method:	MLE	Df Model:	15
Date:	Sat, 27 Apr 2024	Pseudo R-squ.:	0.5336
Time:	09:36:43	Log-Likelihood:	-385.27
converged:	True	LL-Null:	-826.06
Covariance Type:	nonrobust	LLR p-value:	3.088e-178

	coef	std err	z	P>|z|	[0.025	0.975]
Wing_area	-134.3298	32.850	-4.089	0.000	-198.714	-69.945
Wing_shape	44.3095	28.709	1.543	0.123	-11.959	100.578
Wing_vein	17.6097	8.938	1.970	0.049	0.092	35.127
Asymmetry_wing_area	4.7440	6.154	0.771	0.441	-7.318	16.806
Asymmetry_wing_shape	3.0626	3.317	0.923	0.356	-3.439	9.564
Asymmetry_wing_vein	-1.7621	1.456	-1.210	0.226	-4.615	1.091
Thorax_length	-261.0316	77.376	-3.374	0.001	-412.686	-109.377
l2	16.7806	8.892	1.887	0.059	-0.647	34.208
l3p	-173.5110	143.165	-1.212	0.226	-454.110	107.088
l3d	-281.6713	142.214	-1.981	0.048	-560.405	-2.938
lpd	306.5396	168.510	1.819	0.069	-23.734	636.813
l3	33.2353	130.592	0.254	0.799	-222.721	289.191
w1	219.0625	94.476	2.319	0.020	33.893	404.232
w2	19.3733	12.967	1.494	0.135	-6.041	44.787
w3	49.4896	11.996	4.126	0.000	25.978	73.001
wing_loading	-115.9230	47.850	-2.423	0.015	-209.707	-22.139

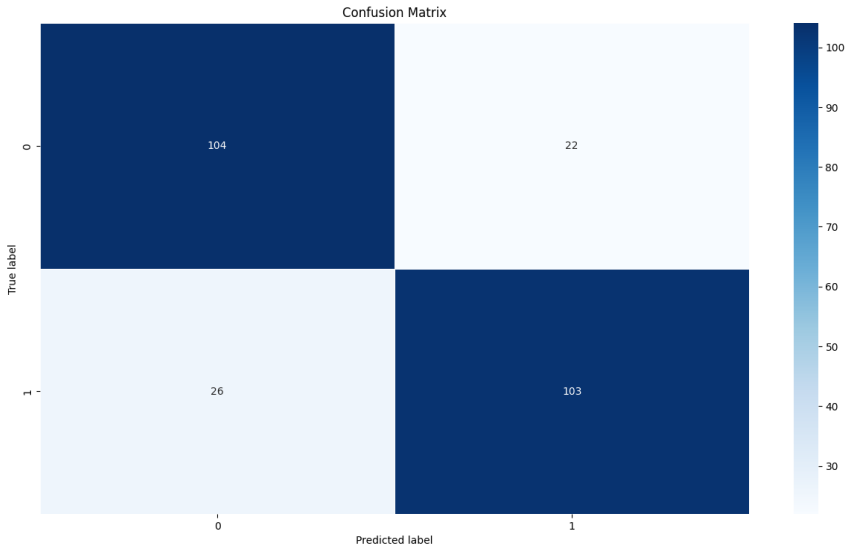
Initially, the model accuracy was 80.08%, which will serve as a baseline accuracy for us to continually improve off of. In order to attempt to improve this predictive capability, we drop the variables 'Wing_shape', 'Asymmetry_wing_area', 'Asymmetry_wing_shape', 'Asymmetry_wing_vein', 'l3p', 'l3', 'w2' as they all fail our hypothesis test at $\alpha = 0.1$. Re-training the model once these variables have been removed results in:

Logit Regression Results

Dep. Variable:	Sex	No. Observations:	1192
Model:	Logit	Df Residuals:	1183
Method:	MLE	Df Model:	8
Date:	Sat, 27 Apr 2024	Pseudo R-squ.:	0.5273
Time:	09:46:01	Log-Likelihood:	-390.46
converged:	True	LL-Null:	-826.06
Covariance Type:	nonrobust	LLR p-value:	9.118e-183

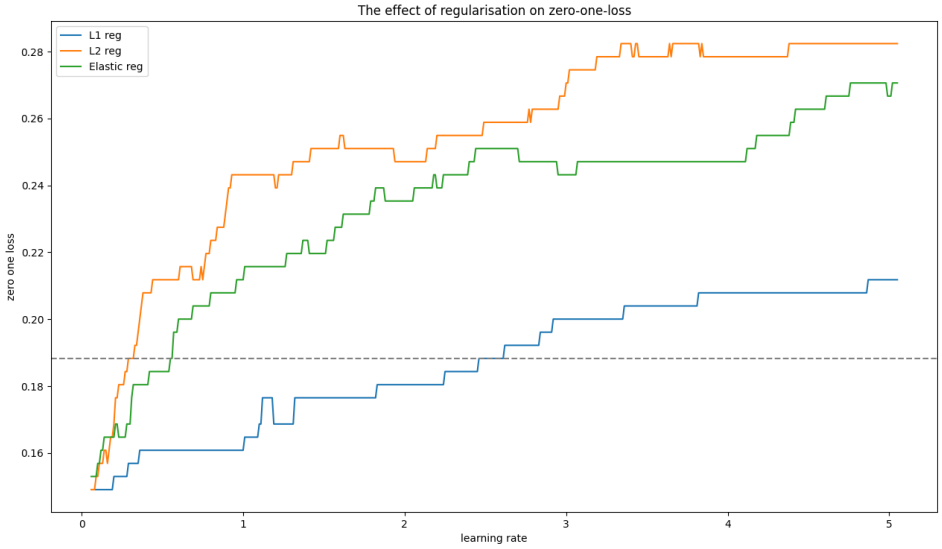
	coef	std err	z	P>|z|	[0.025	0.975]
Wing_area	-102.5290	26.585	-3.857	0.000	-154.634	-50.424
Wing_vein	18.4276	8.455	2.179	0.029	1.855	35.000
Thorax_length	-149.5556	32.635	-4.583	0.000	-213.519	-85.592
l2	28.0118	4.475	6.259	0.000	19.241	36.783
l3d	-111.3200	50.423	-2.208	0.027	-210.147	-12.493
lpd	139.9489	57.372	2.439	0.015	27.502	252.396
w1	104.6141	28.089	3.724	0.000	49.561	159.668
w3	34.1652	6.042	5.655	0.000	22.323	46.007
wing_loading	-46.6963	19.922	-2.344	0.019	-85.742	-7.651

This pruning has marginally increased the predictive power of the model to an accuracy of 81.18%, which is not a significant increase from our original model. Additionally, inspecting of the 95% confidence intervals shows that, for most variables, there is great variance within the least squares parameter estimates. Nevertheless, we are able to infer that variables such as `Thorax_length` and `Wing_area` have a greater impact on the sex of the fruit fly than other variables such as the `Wing_vein`. This is due to the fact that these coefficients have the largest magnitude. Furthermore, to investigate the performance of the algorithm, we present the confusion matrix of the results from the test set:



Our confusion matrix is close the being symmetric, so we can notice that the model has no bias in the prediction of female vs male fruit flies. In addition to this, we have TPR $= 104/(104 + 26) = 0.8$, which is reasonable.

Now that we have derived an optimal set of variables used to predict the sex of the fly in the logistic regression algorithm, we now investigate the affect of regularisation on our predictions. To investigate this, we will conduct a numerical experiment to determine (1) The optimal penalty technique, and (2) The optimal penalty rate. In (1), we will consider ℓ_1, ℓ_2, and elastic regularisation. Let us use $J(\boldsymbol{\theta}; \mathbf{X}, \mathbf{y})$ to define a known cost function in a parametric learning problem given data $(\mathbf{X}, \mathbf{y})$, and unknown parameters $\boldsymbol{\theta}$. We define our new, penalised cost function with regularisation penalty rate $\lambda \in \mathbb{R}^+$ to be $J'(\boldsymbol{\theta}; \mathbf{X}, \mathbf{y}) = J(\boldsymbol{\theta}; \mathbf{X}, \mathbf{y}) + \lambda R(\boldsymbol{\theta})$. The three methods that we investigate amount to choosing different function R and seeking optimal λ. In the ℓ_1 regularisation, we take $R_1(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_1$. In the ℓ_2 regularisation, we take $R_2(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_2^2$. Finally, as not covered in lectures, in our elastic regularisation we take $R(\boldsymbol{\theta} : p) = pR_1(\boldsymbol{\theta}) + (1 - p)R_2(\boldsymbol{\theta})$, where $p \in [0, 1]$ denotes our ℓ_1 ratio. In this experiment, we fix $p = 1/2$ as to have equal weight in our ℓ_1 and ℓ_2 penalties. Now, our numerical experiment will test a set of possible regularisation parameters $\Lambda = \{0.05 + 0.01n : 1 \leq n \leq 500, \in \mathbb{N}\}$, and report the zero-one loss for each. The results of the experiment are:



n.b: The horizontal line is the accuracy of the non-regularised model.

Interestingly, we find that in all three penalty methods, the optimal value for λ is 0.06. In this, both the ℓ_1 and ℓ_2 penalty methods obtain a zero to one loss of ≈ 0.149, whilst the elastic loss results in a slightly higher 0.1529. Based on this, we will select the ℓ_2 penalty method with regularisation parameter $\lambda = 0.06$ as our optimal logistic regression model. Doing so gives us an approximation of our generalisation accuracy at approximately 85.1%, a nice improvement from our non-regularised model.

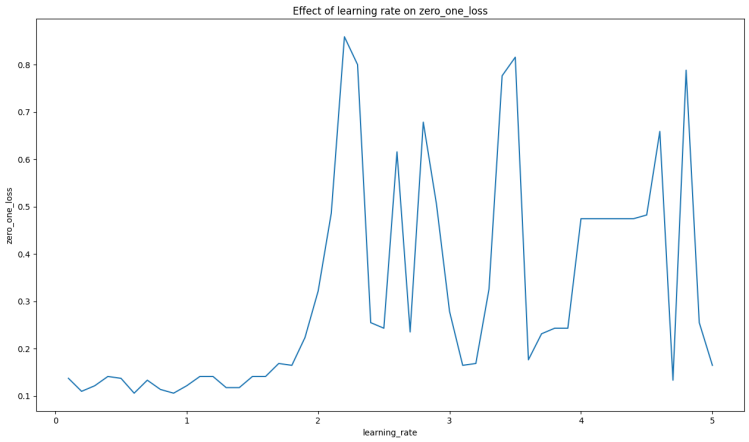
A further analysis of the above graph shows that for ranging λ, the ℓ_1 regularisation is consistently better performing than the other explicit regularisation techniques. ℓ_1 regression has a tendency to restrict our parameter vector $\boldsymbol{\theta}$ to be sparse (i.e. force certain parameters of 0). Despite the resultant cost function being convex in many situations, traditional ℓ_2 regularisation is often faulted for the fact that it does **not** force parameters to zero, as small parameter values have almost no contribution in the regularisation term. Forcing certain parameters to zero in the case of logistic regression allows us to determine which explanatory variables are not relevant to the prediction of the sex of the fly, ultimately yielding a more simple model. So, the fact that ℓ_1 regularisation out performs ℓ_2 is indicative of the fact that

perhaps certain explanatory variables within our data are statistically insignificant in determining the sex of the fly. Based on the `coef` values presented on page 6, we hypothesise that the `Wing_vein` variable is least significant, as it has the smallest coefficient.

Whilst it was nice that we could interpret the coefficients from this model, we sacrifice the interpretability of the results in order for potentially superior predictive powers that can be derived from more complex, non linear models.

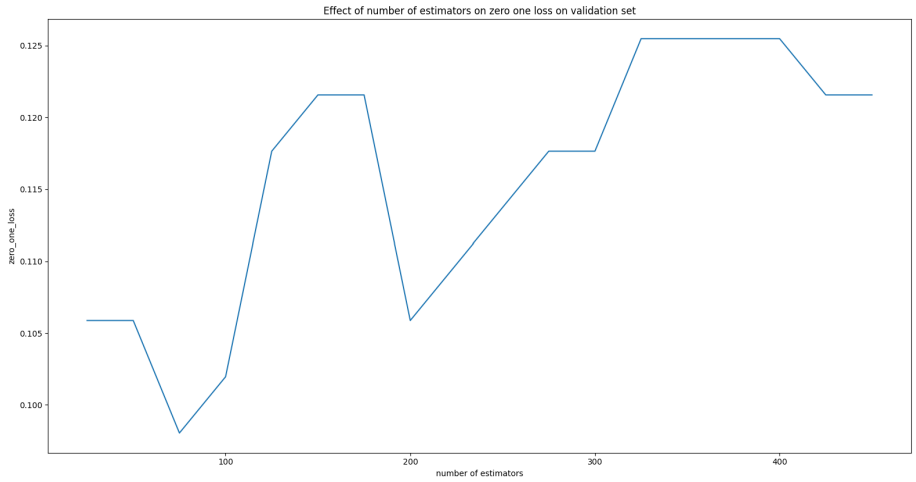
3 Gradient Boosting Tree Classifier

Now, we investigate the use of a gradient boosting random forest classifier in order to predict the sex of the fly. Here, I will use the `sklearn` library to perform the optimisation on my behalf. However, in order to do this we must specify the learning rate, or step size (denoted γ), that we will use in the algorithm. To do this we will define the set $\Gamma = \{0.1 \cdot n : 1 \leq n \leq 50, n \in \mathbb{N}\}$ to be the set of all candidate solutions that we will test. Unlike the the neural network trained in the next section, this model is relatively fast to train so we can afford to test a variety of candidate values. Now, we conduct the numerical experiment in order to find the best $\gamma \in \Gamma$ that we will use in our algorithm. Our experiment will consist of training a model for each $\gamma \in \Gamma$, and evaluating the zero to one loss on the testing set for each model. Plotting the results of this experiment on `matplotlib` result in:



Now, the test reports $\gamma = 0.6$ as the optimal learning rate in our gradient boosting algorithm, which achieves a zero to one loss of 0.106, and consequently an accuracy of 89.4% on the test set. This is a 4.3% improvement from our logistic regression model, which is significant given the restricted nature of the data provided. Interestingly enough, the results from this experiment align with the theory rather nicely! As our learning rate gets large (> 1.5), we see that the convergence of the boosting algorithm become rather chaotic, as the algorithm is now, due to the massive step size, unable to accurately locate some minima on our loss function.

Now that we have fixed some optimal learning rate, we may also investigate how changing the number of boosting stages results affects the performance when fixing our learning rate. In this numerical experiment, we will consider our set of feasible boosting stages to be $\mathcal{B} = \{25 + 25b : 0 \leq b \leq 17\}$. I.e. we consider all boosting stages from 25 to 450 in increments of 25, and record the zero-one-loss on the validation set for models each training with a specific boosting rate $b \in \mathcal{B}$. The results of our experiment are:



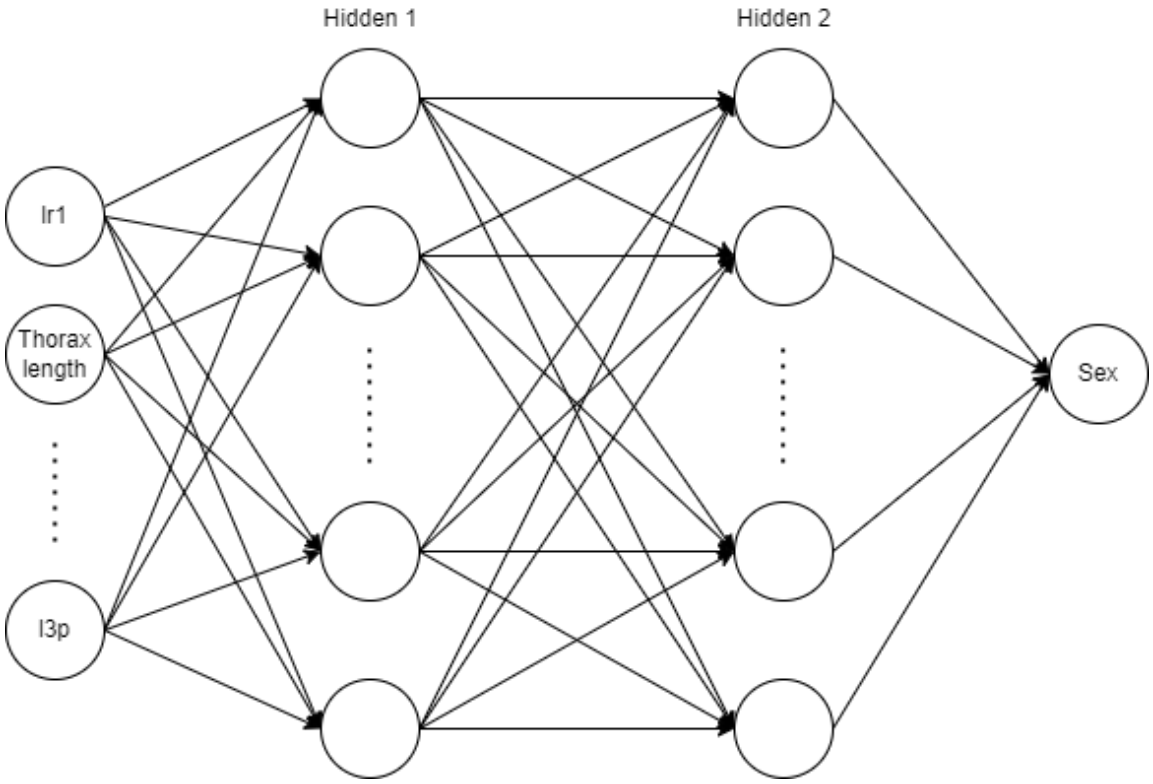
Surprisingly, the classification accuracy reaches a maximum at $b = 75$, which is relatively small. In the scheme of things, we are considering a classification problem of 16 variables, we may simply not need to added complexity. This is additionally helpful because the model is less computationally intensive to train. When taking $b = 75$ and our learning rate at 0.6, we observe an accuracy of 90.2% accuracy on the validation set. This is a welcomed 0.8% improvements from our previous best model. Furthermore, the optimal model produces the following confusion matrix:

$$\text{Confusion matrix} = \begin{pmatrix} 113 & 12 \\ 12 & 117 \end{pmatrix}$$

This is a clear improvement from our logistic regression model, and is representative of the fact that our more complex model is better able to capture complex relationships between the explanatory, and response variables.

4 2-Layer multi layer perceptron

For the final, and also most complex model, we train a feed forward multi layer percepton with two hidden layers. Whilst not covered in COMP4702, I have implemented this model using the open source Deep Learning library PyTorch. The purpose of implementing a more complex, non-linear neural network is to ideally capture complex, non-linear relationships between the explanatory variables, and the response variable. Our model schematic is:

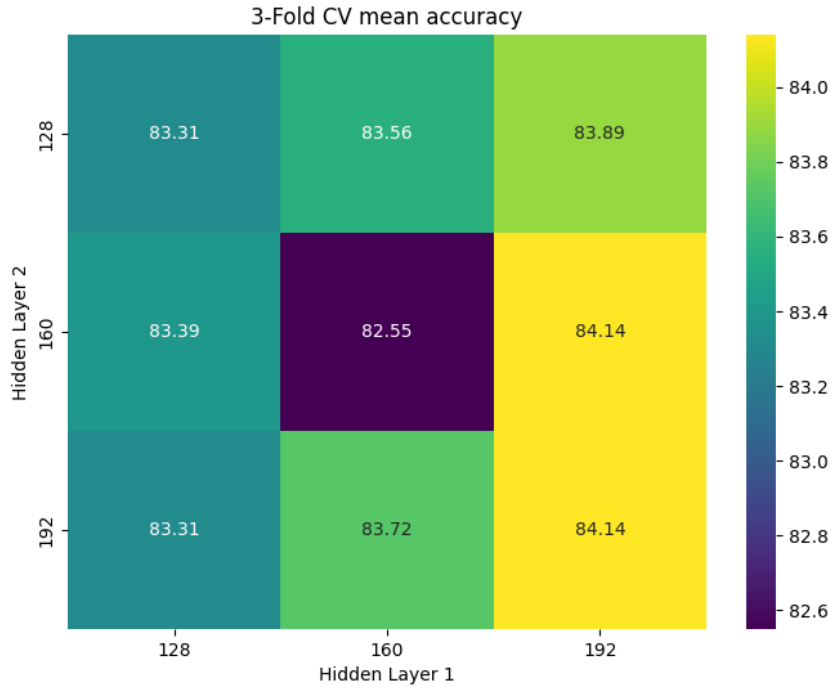


More specifically, the neural network will use the ReLU activation function for the hidden layers, and a sigmoid on the output in order to compress the prediction into the unit interval. In training the model, we use binary cross entropy (BCE) loss. Given a prediction, $\hat{y}$, and a ground truth value p, we define:

$$\text{BCE}(\hat{y}, p) = -(\hat{y}\log(p) + (1-\hat{y})\log(1-p))$$

Furthermore, we will use the Adam optimizer with predefined learning rate 0.001.

Currently, the dimension of the two hidden layers is unknown, and will be found through hyper-parameter tuning performed in the testing. Specifically, to determine the dimension of hidden layers 1 and 2, which we will denote d_1 and d_2, we will use 3-Fold cross validation on the training set. We will induce the inductive bias that $d_1, d_2 \in \{128, 160, 192\}$, as there are hardware restrictions on the laptop in which the model was trained on. The results from this cross validation testing are:

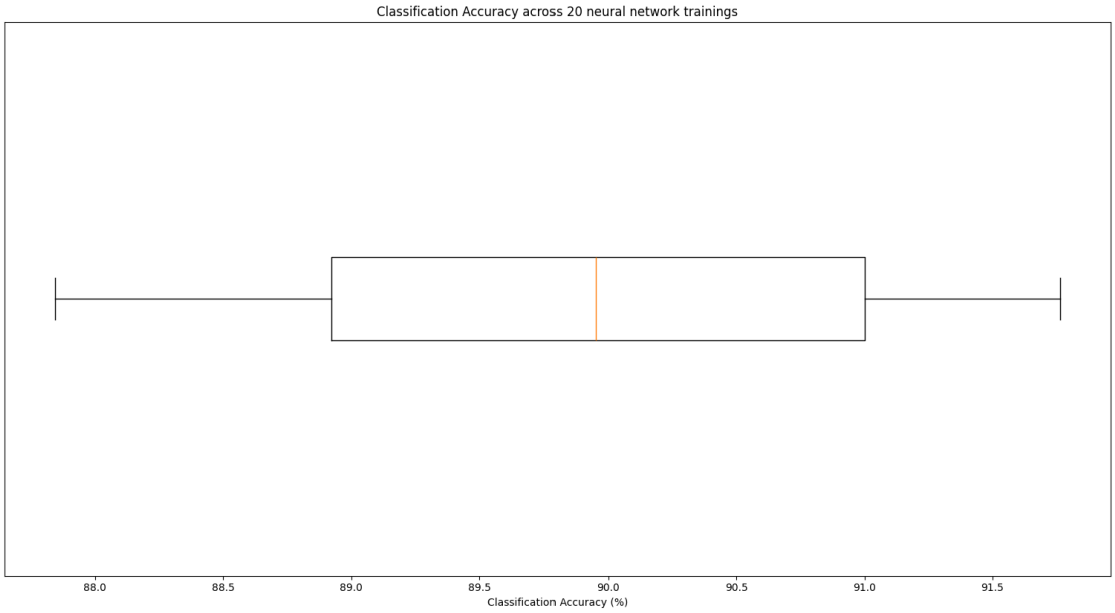


Thus, either $d_1 = d_2 = 160$ or $d_1 = 192, d_2 = 160$ are our optimal hidden layeer dimensions. Actually, the performance when $d_1 = d_2 = 160$ is marginally better than the alternative, and they appear identical on the above image due to truncating in the floating point values. Taking these specification and re-training the resulting model on the training set, and then tested on the unseen testing set provides the estimation of the generalisation accuracy at 91.76%. This is a marginal improvement on our existing performance. It is important to note also that this is significantly higher than our 3-Fold mean CV accuracy portrayed in the above grid. This is likely due to the fact that in the 3-Fold CV training process, the model

is being trained on a restricted dataset, so is unable to capture the relationship dynamics. Once the model is trained on more data, it's accuracy increases. We can investigate this further by looking at the associated confusion matrix:

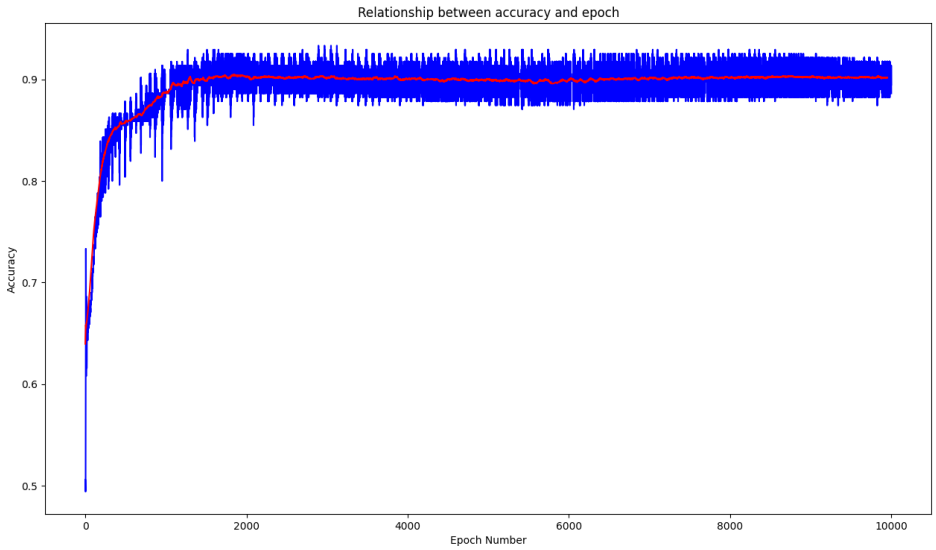
$$\text{confusion matrix} = \begin{pmatrix} 115 & 11 \\ 13 & 136 \end{pmatrix}$$

As the confusion matrix is almost symmetric, our model does not over-predict either sex, and for can accurately identify the sex of the fly based purely off quantitative measurements of it's features. However, the training of neural Networks is indeed stochastic, so to fully investigate the effectiveness of the neural network model, we should investigate the variability within our training accuracy. To investigate this, we conduct 20 separate training periods, and report the accuracy on the validation dataset for each:



More specifically, we find that our mean prediction accuracy is 89.92%, and the standard deviation within the data is 1.14 units. The top end of this is an improvement on our previous models. Now that we have decided on the model dimensions, we can further investigate the number of epochs. Previously, all models were trained on 10^4 epochs. Whilst large, the models did not experience over fit-

ting, so this was ok. To further investigate this, we have conducted a numerical experiment that plots the accuracy on the test test against the epoch number of our feed forward neural network:



Unsurprisingly, we see that as our epoch number becomes larger than approximately 2000, the accuracy of our model ceases to increase. This suggests that training beyond ≈ 2000 epochs is a waste of compute resources, so we will fix the number of epochs at 2000, concluding our model development phase.

5 Conclusion & Model selection

Now that we have developed three models of increasing complexity, we can ultimately present both a model, and an estimation for the generalisation error of the model to new, unseen data. In order to do this, we present the pros and cons of each model:

1. Logistic regression with ℓ_2 regularisation.

   Pros: Simple, lightweight model with coefficients that reveal insights about the classification problem.

   Cons: Relatively poor predictive power relative to the other classifiers.

2. Gradient boosting tree classifier

   Pros: Faster to train and more lightweight than the neural network.

   Cons: Classification accuracy is *marginally* worse than the neural network.

3. 2-Layer multi layer perceptron

   Pros: Superior predictive power compared to the other two models. Once trained the model it is very fast to predict the outcome.

   Cons: Takes a longer period of time to train the model, and more expensive compute required.

Based on this analysis, we've opted to select the 2-Layer multi layer perception as our 'best' predictive model. As a final section of the report, we provide an estimation of the generalisation error on new, previously unseen data. To do this, we will test the model on the $\approx 15\%$ of the original data that has been held out. Doing this, we find that the model accuracy is 91.67%. This is marginally worse than our accuracy on the validation set, which is representative the the model isn't severely over fitting to the training data. This is something that we have to give great caution to when training such a high variance model like a neural network. Thus, this is our estimation for the accuracy of the model to new, previously unseen data. Finally, the fact that the final, most complex model only achieves a marginal improvement from our gradient boosting classifier suggests that perhaps the data used in the analysis cannot be used to predict the sex of the fly 100% of the time.