Stephen Bertwistle (30115330)

COMP4702 Assignment

Semester 1, 2024

Chapter 1 Introduction

Loeschcke, Bundgaard and Barker (2000) published a study of the fruit fly species *Drosophila aldrichi* and *D. buzzatii*. Students from COMP4702 were provided several files which represented the raw data for this paper. The brief for this assignment was simple (and open-ended): use some of the Machine Learning techniques on the raw data from Loeschcke *et al*.

My assignment concentrates on the data summarised in the file 83_Loeschcke_et_al_2000_Thorax_&_wing_traits_lab pops.csv. This relates to the physical measurements of approximately 1700 individuals divided roughly equally between the two species and between male and female. Therefore, there are two obvious binary classification problems which could be studied. Is it possible to build machine learning models to predict the sex of an individual? Is it possible to build models to predict the species of an individual? Another more complication classification problem would be a model to predict sex and species simultaneously.

In any scientific enquiry, it often best to start with the simplest experiments first. For this reason, I chose to start with two simple machine learning techniques to address these classification problems, namely, k-Nearest Neighbours (k-NN) and Logistic Regression. If these techniques provide adequate results, it may not be necessary to apply more complicated methods such as neural networks.

Chapter 2 Materials and Methods

The coding for this assignment was performed using the Python 3.11.5 programming language with publicly available libraries and classes.
The libraries/classes were -

- numpy
- pandas
- math
- matlibplot
- sklearn.metrics.confusion_matrix
- sklearn.metrics.ConfusionMatrixDisplay
- sklearn.model_selection.test_train_split
- sklearn.neighbours.KneighborsClassifier
- sklearn.linear_model.LogisticRegression
- scipy.integrate.trapezoid

The development environment employed was Spyder 5.4.3.

The datafile provided for this assignment was 83_Loeschcke_et_al_2000_Thorax_&_wing_traits_lab pops.csv which contained raw data for a paper published by Loeschcke *et al*.

Several columns in the dataset did not relate to the physical characteristics of the individual fruit flies and were discarded. These columns were -

- Population
- Latitude
- Longitude
- Year_start
- Year_end
- Temperature
- Vial
- Replicate

Rows where wing_loading or Thorax_length was not a number were discarded. As were rows where l2 was zero.

Data in every numeric column was normalised so that each column ranged from 0 to 1. This is important for equal weighting of the columns in k-NN models.

Sex was converted to a numeric value (0 for male, 1 for female). For the 4-way classification, species and sex were combined to a numeric value (0 for D. aldrichi male, 1 for D. aldrichi female, 2 for D. buzzatii male, 3 for D. buzzatii female).

The rows in the dataframe were randomly shuffled using a seed value. Then 30% of the rows were set aside as the test dataset. Thus “test results” relate the same dataset in every experiment.

Area Under Curve (AUC) was estimated using trapezoidal integration.

Chapter 3 k-Nearest Neigbours

3.1 Prediction of the fly's sex

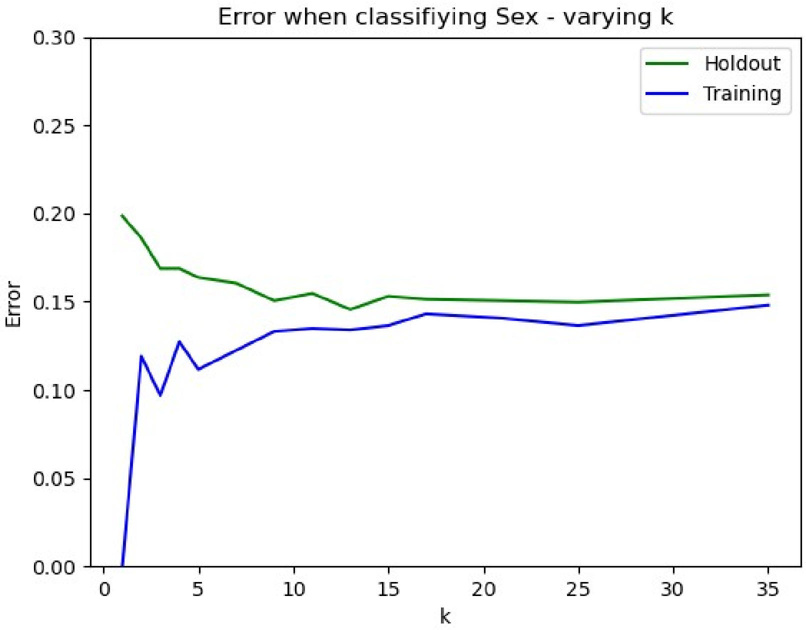
3.1.1 Selection of k

In this experiment, the training dataset was divided into 5 folds. (I will not use the term "k-fold validation" to avoid confusion with "k-nearest neighbours".) The following algorithm was employed -

```
For each value of k -
      For each of the 5 folds -
            Set that fold aside as holdout data.
            Train the k-NN model on the remaining 4 folds.
            Calculate the error on the holdout data.
      Calculate the mean error of the holdout data for that k.
      Train the model again for all training data using that value of k.
      Calculate the error across all of the training data for that k.
```

In this way, we can assume E-holdout is a reasonable estimation of E-new, that is the error rate we might expect when the model is applied to new "production" data.
The results have been summarised in Figure 3.1.1.

Figure 3.1.1



It is interesting to compare Figure 3.1.1 to Figure 4.3 from Lindholm, Wahlstrom, Lindsten and Schon (2022) (reproduced in Appendix A). First note that the direction of the horizontal axis has been reversed. Increasing values of k are believed to represent decreasing model complexity as the model becomes less sensitive to noise in the training data. It can be seen that the generalisation gap (the difference between E-holdout and E-training) is greatest for lower values of k (greater model complexity).

As expected, E-training is zero for k=1. Each datapoint in the training set is its own nearest neighbour. Testing the model against the training set for k=1 will produce no classification errors. E-holdout is also greatest when k=1 which is to be expected as this when the model is most complex and will not generalise well to unseen data.

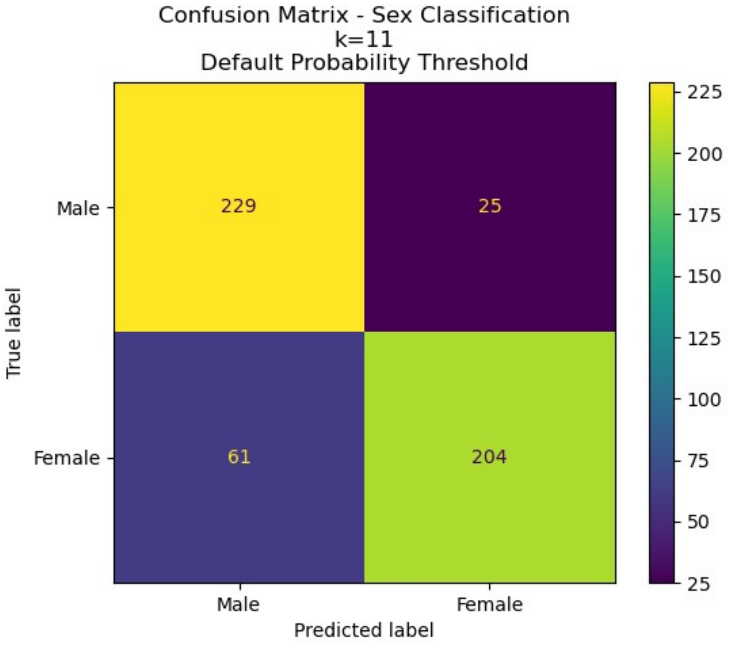
The generalisation gap narrows for k>10 but never completely closes which again agrees with the idealised graph from Lindholm *et al*.

When using k-NN for binary classification, the model will calculate the Euclidean distance between a new datapoint and all datapoints in the training set. If the majority of the nearest neighbours are of male (since here were attempting to predict the sex of an individual), the model will output “male”. Otherwise “female” will be returned. To avoid ties (which might require some random tie-breaker), it is best to select an odd number for k.

For these reasons, the value of k=11 was chosen for future experiments when attempting to predict the sex of a fruit fly using the k-NN technique.

3.1.2 Prediction of fly’s sex using knn – Confusion Matrix

Figure 3.1.2



Error for Training Data: 0.135
Error (Misclassification Rate) for Test Data: 0.166
Recall (True Positive Rate - Male): 0.902
Precision (Male): 0.790
False Positive Rate (Male): 0.230

The confusion matrix in Figure 3.1.2 represents the results of testing the k=11 model against the test dataset, that is, data which was not used in training the model. (This applies to all confusion matrices displayed in this report.)

For the purposes of this matrix, “male” was considered to be positive and “female” negative. The accuracy was calculated as the rate at which sex was correctly predicted. That is, the sum of the entries on the main diagonal divided by the total number of individuals. Error rate was the misclassification rate, that is 1 – accuracy. Recall (as known as True Positive Rate), Precision and

False Positive Rate were calculated as described in Table 4.1 of Lindholm *et al* (reproduced in Appendix A).

The Error rate of 0.166 agrees quite well with the E-holdout value of approximately 0.15 which can be read from Figure 3.1.1. This gives us some confidence that selecting k=11 was a reasonable choice and that E-new for unseen data will be similar.

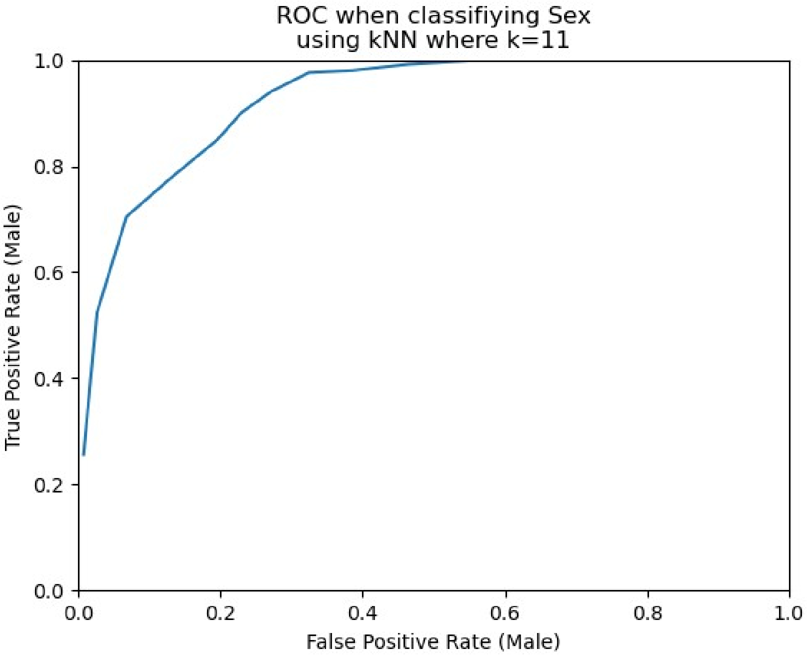
E-train (0.135) is slightly better than E-test (0.166). This is to be expected. But the generalisation gap is not particularly large.

While the TPR is quite high (greater than 0.9), the FPR (0.23) is perhaps uncomfortably high as well.

3.1.3 – Prediction of fly's sex using knn – ROC and Precision/Recall

The confusion matrix in Figure 3.1.2 employs the default probability threshold of $r = 0.5$. The k-NN model produces a probability for its prediction based on the proportion of the nearest neighbours which are of a particular sex. For example, if 6 of the nearest neighbours are male, the model calculates the probability of that individual being male as 6/11. If this probability is greater than or equal to the threshold r, then the model outputs "male", otherwise it outputs "female". Thus the characteristics of the model (TPR, FPR, recall, precision) can be altered by varying r.

Figure 3.1.3.1

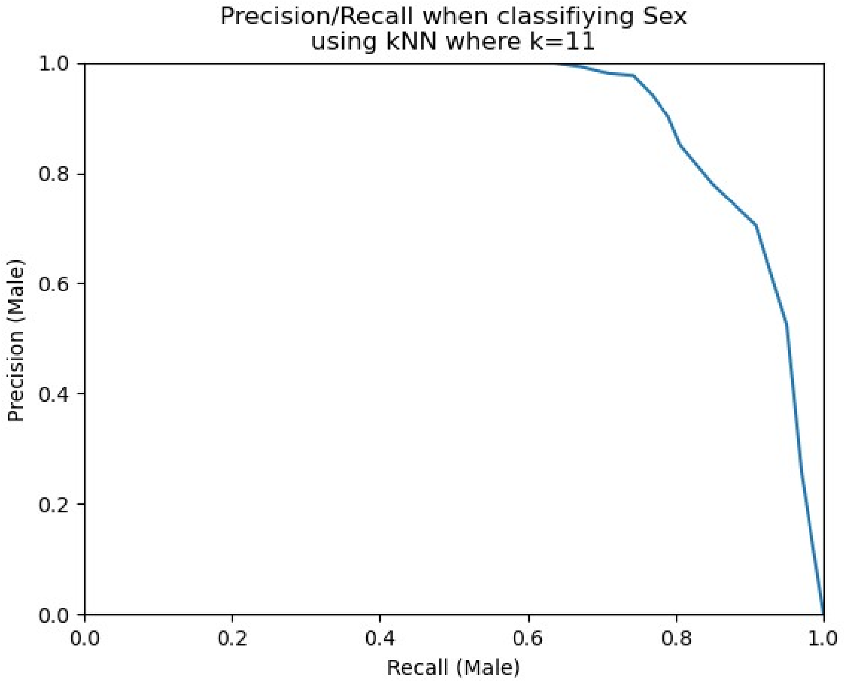


This graph was created by varying the probability threshold r, from 1 to 0 and calculating the True Positive Rate (TPR) and False Positive Rate (FPR) each time. Moving from right to left along the graph represents decreasing r. (k remained constant at 11.)
It can be seen that there is a trade-off: decreasing r results in increasing TPR, but as expected, FPR also increases.

Let us suppose that the male fruit fly is more dangerous to crops than the female. Then the model needs to identify as many males as possible. It can do this by increasing *r* above 0.5. This comes at a cost however. It will increase the proportion of females incorrectly identified as males.

The Area Under Curve (AUC) was estimated using trapezoidal rule as 0.924. A perfect model will have AUC of 1. (Compare to the Lindholm's Figure 4.13a reproduced in Appendix A.) While a model employing random guessing will have an AUC of 0.5. We can therefore conclude that this model is much closer to a theoretically perfect classifier than it is to a random classifier.

Figure 3.1.3.2



The graph of Recall versus Precision is produced here mainly for completeness. The data in the test and training contains roughly equal number of males and females. That is, it is balanced. Lindholm *et al* suggest such a graph may be more useful for imbalanced data. Still Figure 3.1.3.2 is roughly the same as theoretical graph produced by those authors as Figure 4.13a (again re-produced in Appendix A).

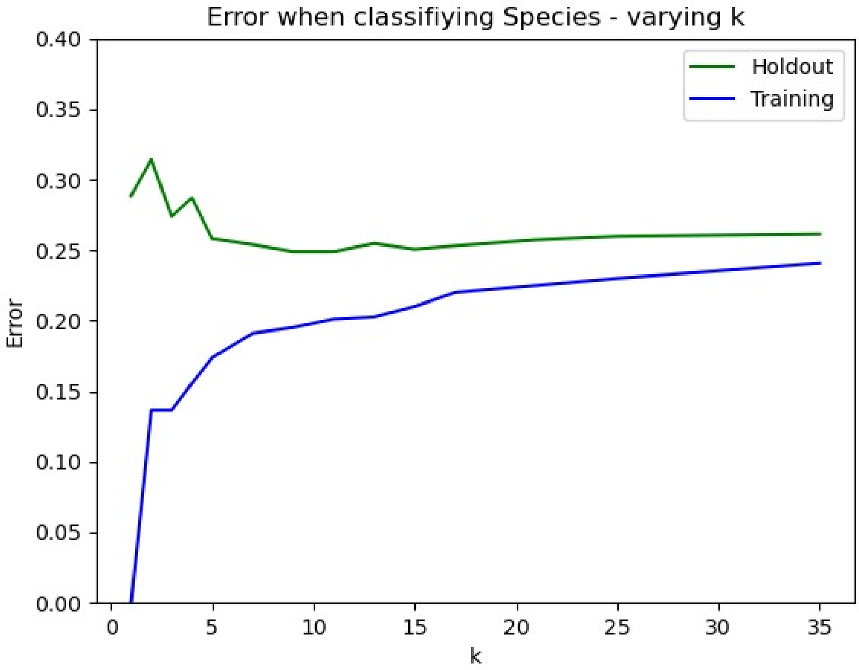
The AUC was calculated as 0.916. Again, much closer to a theoretically perfect classifier (1.0) than random guessing (which would have an AUC of 0.5 in this case since the data is balanced).

3.2 Prediction of Fly's Species

From the data supplied in the CSV file, another obvious binary classifier would be to predict the species of the fly (*D. aldrichi* or *D. buzzatii*) using k-NN.

3.2.1 Selection of k

Figure 3.2.1



The experiment to select k for species classification was similar to that described in Section 3.1.1 (for sex classification). The results are summarised in Figure 3.2.1.

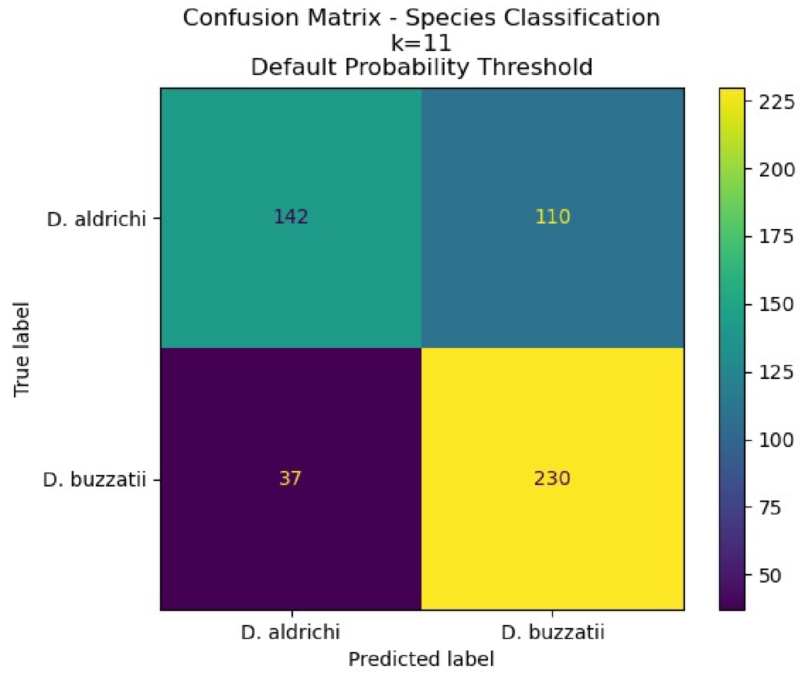
Again we see that the generalisation gap is greatest for lower values of k (greater model complexity). The gap decreases but does not close entirely as k increases (model complexity decreases). As expected, the E-train is zero for k=1.

E-holdout appears to be a minimum at values near k=10, 11 or 12. However, the minimum E-holdout is 0.25 which is significantly higher than for sex classification (which was 0.15).

Again k=11 was selected for species classification.

3.2.2 Prediction of fly's species using k-NN

Figure 3.2.2



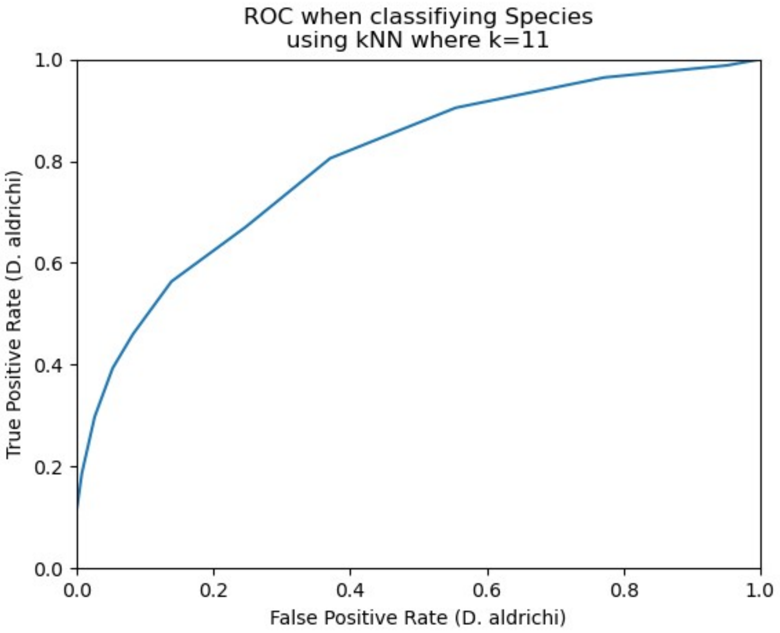
Test Error (Misclassification) Rate: 0.283
Recall (True Positive Rate): 0.563
Precision: 0.793
False Positive Rate: 0.139

The confusion matrix in Figure 3.2.2 presents the *D. aldrichi* as the "positive" class.
The E-test (0.28) is slightly above the E-holdout (0.25) from section 3.2.1. We can usually expect E-new to be higher than E-holdout. The fact that they are so close gives us confidence that the error for future "unseen" data will be similar. E-train was 0.201. A relatively small generalisation gap suggests that the model has not been over-fitted to the training data.

The TPR is quite low at 0.56. That is, a little over half of the actual *D. alrichi* specimens are correctly identified as such by the model. Balancing this is the low False Positive Rate (0.14). This suggests that the model may benefit from decreasing *r* to below 0.5 if the objective is to identify as many *D. aldrichi* individuals as possible.

3.2.3 Prediction of Species – ROC and Precision/Recall

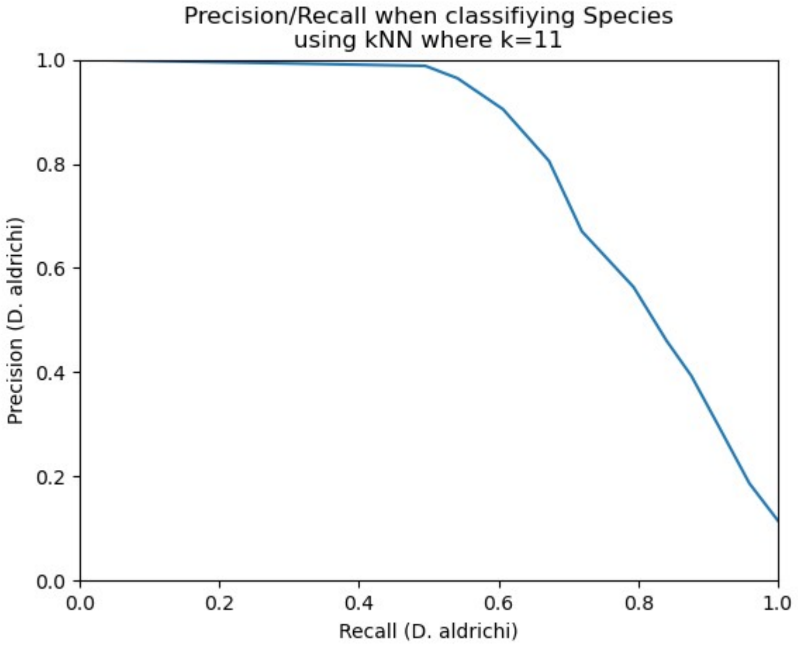
Figure 3.2.3.1



Moving along the graph from right to left represents decreasing the probability threshold, *r*, from 1 to 0. It can be seen that to achieve a TPR of 0.9, we must endure FPR of 0.6 or higher. This reinforces the point that predicting species using k-NN is not as successful as predicting sex.

AUC was calculated to be 0.798. This is still closer to the theoretical perfect classifier (AUC=1) than random guessing (AUC=0.5).

Figure 3.2.3.2

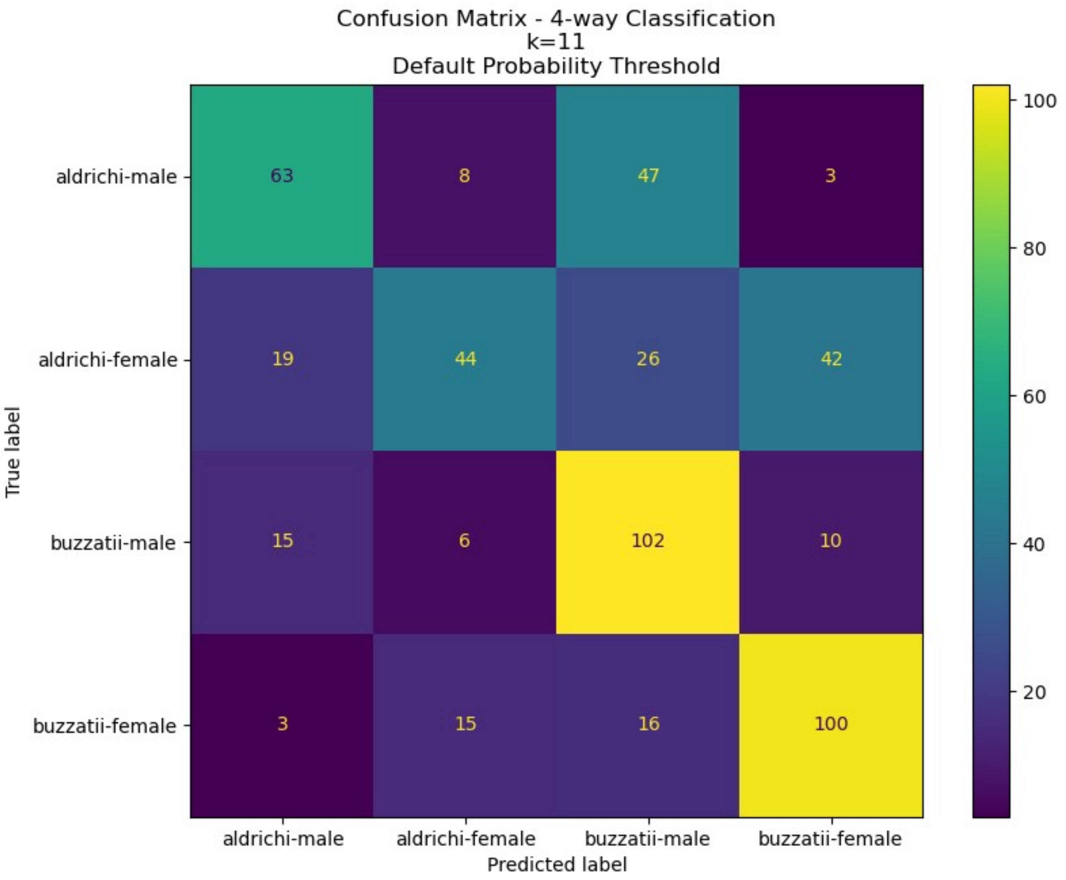


For completeness, the Precision/Recall graph is displayed here. AUC was calculated to be 0.804.

3.3.1 Four-class Classification using k-NN

k-NN was employed to determine if it can successfully predict a four-way classification (*D. aldrichi*-male, *D. aldrichi*-female, *D. buzzatii*-male, *D. buzzatii*-female)

Figure 3.3.1



The test error rate for the 4-way classification was 0.405. This is significantly higher than either of the binary classifiers using k-NN (0.17 for sex, 0.28 for species). However, if the model was simply randomly guessing, we might expect an error rate of 0.75, that is, an accuracy rate of 1 in 4. Admittedly, comparing any model to a random model is a very “low bar to clear”.

Consider that the accuracy rate of the 4-way classification of approximately 0.6. Then assume that the classification of sex is independent of the classification of species. We can then multiply probabilities of correctly calling the each of the binary classifications.
That is 0.83 multiplied by 0.72 is approximately 0.60, which is close to the observed accuracy of the 4-way classifier. In this sense, the 4-way classifier is as accurate as to be expected.

The 4-way model does appear to identify *D. buzzatii*-male and *D.buzzatii*-female quite well. If we consider the bottom row of the confusion matrix, 100 of the 134 of *D. buzzatii* females were correctly identified. That is a recall of 0.746. This appears to be at the expense of over predicting this class. If we consider the rightmost column, the model predicts 153 *D. buzzatii* females of which only 100 are correct. That is a precision of 0.653. However, if we consider the top row, the model correctly predicts 63 of the 121 actual *D. aldrichi* males (recall of 0.521). Using the leftmost column, the model predicts 100 *D. aldrichi* males of which only 60 are correct (precision of 0.600).

Chapter 4 - Logistic Regression

Now we will repeat the analysis of Chapter 3 using Logistic Regression rather k-NN in an attempt to determine if it (logistic regression) provides any improvement.

4.1 Prediction of the fly's sex

4.1.1

Logistic Regression was performed using Sklearn's LogisticRegression class. By default, this class uses regularisation. However, the type of regularisation available depends on the solver (numerical optimiser) chosen to do the regression. It was decided to use the 'lbfgs' and 'liblinear' solvers.

Lbfgs is a gradient descent algorithm. In Sklearn, the lbfgs solver allows the choice of L2 regularisation or no regularisation at all.

Liblinear is co-ordinate descent algorithm. In Sklearn, the liblinear solver allows the choice of L1 or L2 regulatisation. It appears that it does not offer the option of switching off regularisation.

For this experiment, the regularisation parameter was left as its default value. Time constraints (and the length of this report) did not allow experimenting with this parameter. The results are summarised in Table 4.1.1.

Table 4.1.1 – Logistic Regression using different solvers and regularisation

solver	lbfgs		liblinear			
regularisation	none	L2	L1	L2	L1	L1
Features	all	all	all	all	thorax length, l2, l3d, w3	thorax length
Training Error	0.147	0.180	0.169	0.176	0.169	0.234
Test Results						
error	0.162	0.177	0.152	0.168	0.152	0.218
recall	0.862	0.850	0.866	0.862	0.866	0.791
precision	0.817	0.800	0.830	0.808	0.830	0.770
false positive rate	0.185	0.204	0.170	0.196	0.170	0.226
Model Parameters						
intercept	1.630	-2.940	-7.286	-2.626	-7.287	-5.942
coefficients						
thorax length	-25.650	6.935	22.771	6.476	22.767	11.122
l2	-25.257	-0.604	-5.433	-0.490	-5.439	
l3p	48.009	0.510	0.000	0.495		
l3d	105.991	2.256	1.847	2.185	1.867	
lpd	-30.799	1.910	0.000	1.970		
l3	-19.482	1.898	0.000	1.959		
w1	-3.185	-0.125	0.000	-0.136		
w2	4.509	0.791	0.000	0.887		
w3	-11.052	-2.419	-6.951	-2.343	-6.962	
wing loading	-42.857	-5.881	0.000	-6.263		

First we will consider the results from the lbfgs solver with and without L2 regularisation. Including L2 regularisation results in a slight increase in the error rate. Of greater interest are the parameters of the two different learned models. L2 regularisation is believed to favour a model where the coefficients have smaller absolute values. Indeed this is what we observe here. When regularisation is not employed, the absolute value of the coefficients varies from 3.1 to approximately 106. When L2 regularisation is used, the absolute value of the coefficients varies from 0.125 to 6.9, although none are driven to zero (which is to be expected).

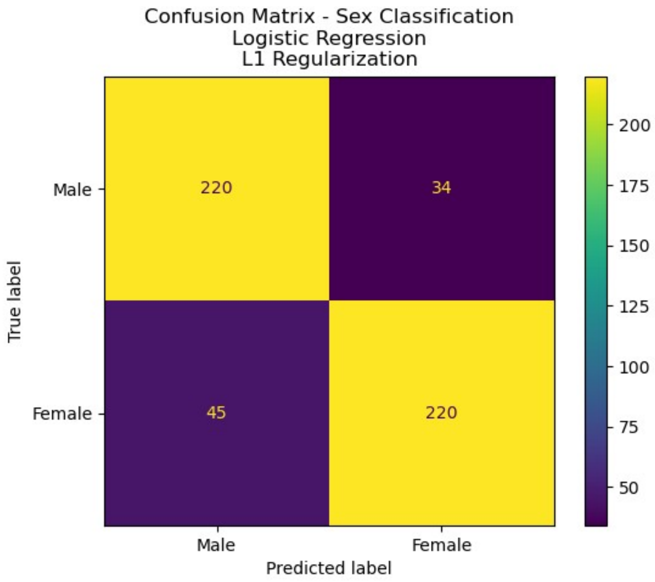
Next consider the results from the liblinear solver using L2 and L1 regularisation and employing all input features. L1 results in a slightly smaller error rate (0.152 compared to 0.168). Again, the greater interest is the coefficients of the learned models. One advantage of using L1 over L2 is that L1 might drive some of the coefficients to zero (“feature selection for free”). That is exactly what we observe here. The coefficients corresponding to l3p, lpd, l3, w1, w2 and wing loading are all zero. This suggests that these input features are not considered important by model produced using L1 regularisation. Indeed, it suggests that these features may be left out all together.

Now consider the results summarised in the column second from right of Table 4.1.1. The model was re-trained using thorax length, l2, l3d and w3 as input features (and using liblinear and L1 regularisation). The test results (with respect to error, recall, precision, FPR) are identical to those obtained using all input features. The coefficients of the model are almost the same as the non-zero coefficients of the model which used all input features. The minor variations could be explained by the random nature of the numerical optimisation algorithm.

Of greater interest is the fact that the coefficient for thorax length is significantly greater than those for l2, l3d and w3. That is, significantly more weight is given to the thorax length. This suggests that an even simpler model may be viable using thorax length as the sole input feature. The results of such a model are summarised in the rightmost column of Table 4.1.1. The error rate has increased from 0.152 to 0.218. This may be significant but it is not disastrous. The simple model using only thorax length may a viable way of predicting the sex of a fruit fly.

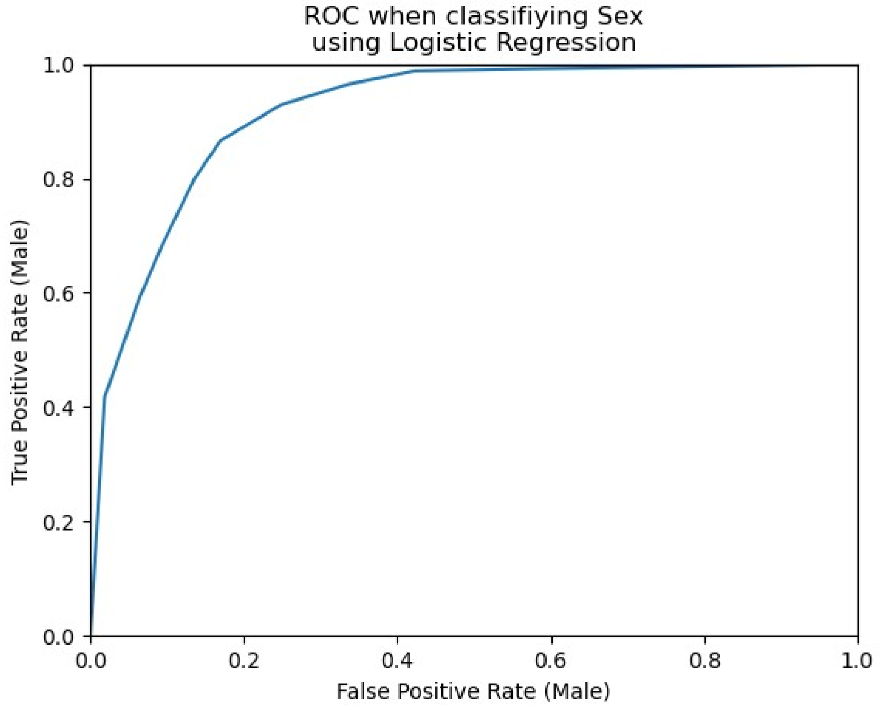
4.1.2 Confusion Matrix for Classification of Sex using Logistic Regression

Figure 4.1.2
This confusion matrix was produced using L1 regularisation and all input features. An identical matrix was produced when the input features were restricted to thorax length, l2, l3d and w3. This again reinforces the idea that L1 regularisation results in “feature selection for free”.



4.1.3 ROC curve and Precision/Recall

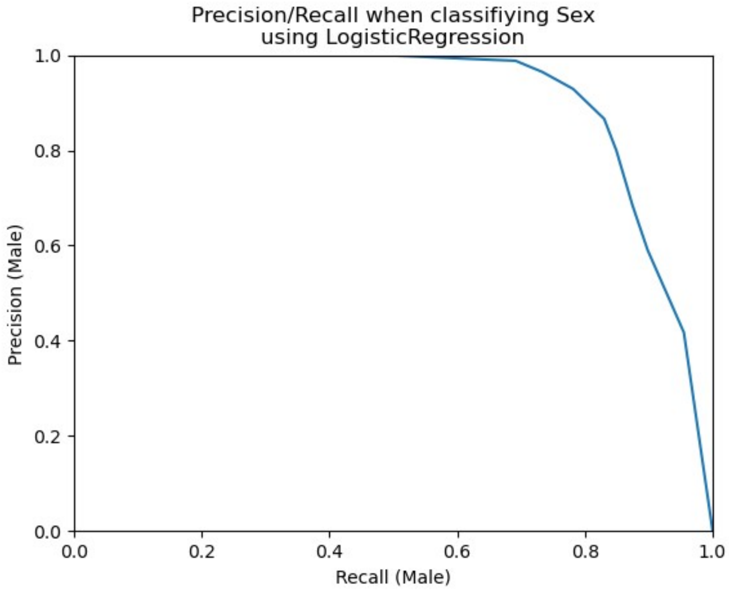
Figure 4.1.3.1



As for the analogous curve using k-NN, moving from left to right represents decreasing *r*, the threshold probability for predicting a male. We can see that increasing the TPR to above 0.9 will mean that the FPR increases to approximately 0.25.
An AUC value of 0.917 suggests that the model is significantly closer to a theoretical perfect classifier (AUC=1.0) than random guessing (AUC=0.5).

For completeness, the Precision/Recall graph has been produced, even though the class division (male/female) is roughly equal.

Figure 4.1.3.2



Area UnderCurve: 0.908

4.2 Prediction of Fly’s Species using Linear Regression

4.2.1
In manner similar to that described in Section 4.1.1, Sklearn’s LogisticRegression was used to determine if a model could be built to predict the species of an individual fly. Again the lbfgs and liblinear solvers were used for numerical optimisation. And again, lbfgs was employed with L2 regularisation and without any regularisation. Liblinear was employed with L1 and then L2 regularisation. The results and model parameters are summarised in Table 4.2.1.

Table 4.2.1 Prediction of Species using Linear Regression

Solver	lbfgs		liblinear			
Regularisation	none	L2	L1	L2	L1	L1
Features	all	all	all	all	sex, l2, l3p, l3d, w1, w2, wing length	l2, w1, w2
Training Error	0.223	0.257	0.228	0.254	0.228	0.249
Test Results						
error	0.208	0.250	0.212	0.249	0.212	0.252
recall	0.754	0.706	0.746	0.706	0.746	0.702
precision	0.805	0.761	0.803	0.764	0.803	0.760
false positive rate	0.172	0.210	0.172	0.206	0.172	0.210
Model Parameters						
intercept	-1.379	-0.690	0.362	-0.630	0.362	-0.247
coefficients						
sex	-0.657	-0.600	-0.665	-0.599	-0.665	
thorax length	21.203	1.274	0.000	1.176		
l2	-9.750	5.417	11.617	5.443	11.612	9.594
l3p	-23.118	1.541	0.443	1.540	0.445	
l3d	-62.912	0.043	-2.706	0.032	-2.702	
lpd	-79.321	0.591	0.000	0.611		
l3	125.808	0.654	0.000	0.674		
w1	-57.055	-6.475	-20.600	-6.477	-20.603	-20.350
w2	49.942	2.461	13.556	2.487	13.559	11.642
w3	25.962	-3.255	0.000	-3.237		
wing loading	14.480	-0.430	-1.543	-0.529	-1.543	

If we first consider the test results we see that lbfgs without regularisation slightly outperforms lbfgs with L2 regularisation and liblinear with L2 regularisation. The error rate for lbfgs (with no regularisation) is very similar to that of liblinear-L1, that is, 0.208 compared to 0.212. The obvious conclusion is that here, regularisation provides little benefit other than the reduction in coefficient size and the “feature selection for free” of L1 regularisation.

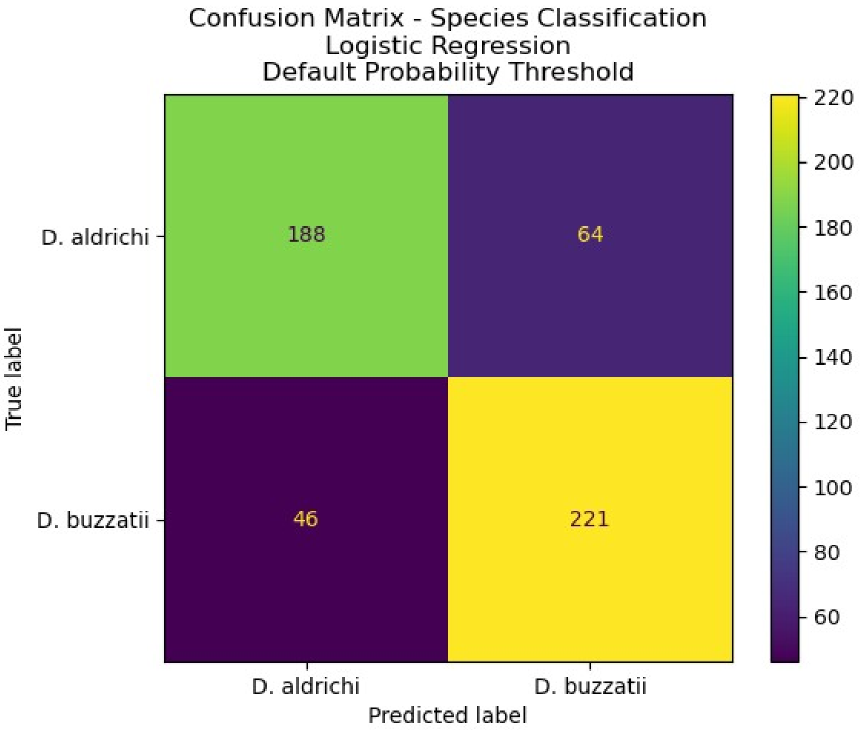
Again, we see that L1 regularisation has driven a number of parameters to zero. These correspond to thorax length, lpd, l3 and w3. This is interesting since thorax length was the major feature in predicting the sex of a fly. Here, after building a model to predict species, L1 regularisation suggests that it (thorax length) is not worth considering.

The liblinear-L1 experiment was repeated with input features sex l2, l3p, l3d, w1, w2 and wing loading. The test results (error, precision, recall, FPR) were identical to those when uses all input feature with liblinear-L1. There were minor variations in trained model parameters.

Another observation is the relatively large absolute value of coefficients corresponding to l2, w1 and w2 in the liblinear-L1 model. This suggests that a simplified model using only these 3 features may be viable. The experiment was repeated by training the model with only l2, w1 and w2 as input features. The error rate increased to 0.252. This indicates that the simplified model with limited input features may still be a viable model.

4.2.2 Confusion Matrix – Prediction of Species by Logistic Regression

Figure 4.2.2

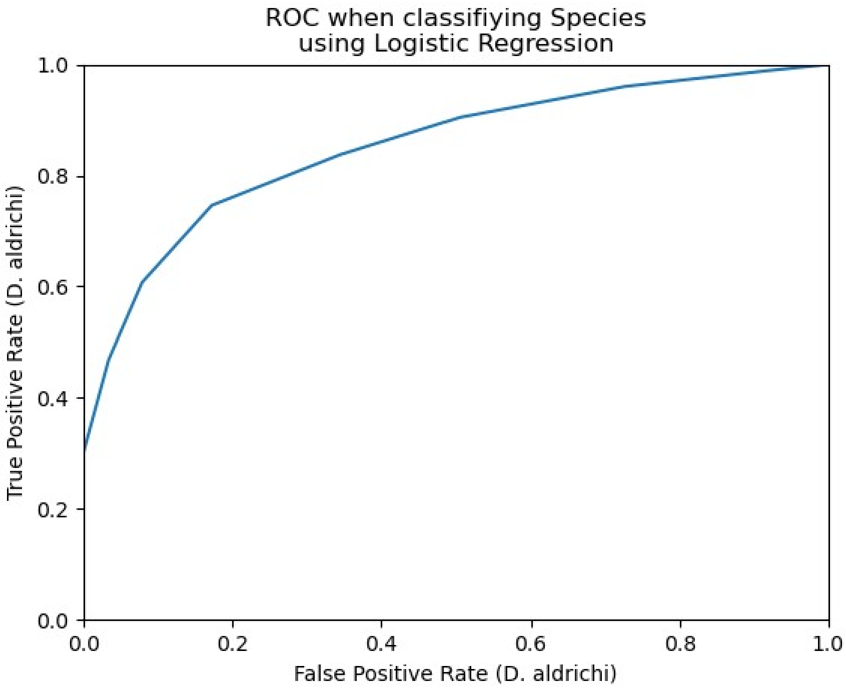


This confusion matrix was produced using the liblinear solver with L1 regularisation (as was Figures 4.3.2.1 and 4.3.2.2). The error rate of 0.212 is significantly higher than that when linear regression was used to predict sex (0.15). This may be due to that fact that thorax length was a strong predictor of sex. There appears to be no feature which predicts species as strongly.

However the error rate for logistic regression predicting species does appear to be an improvement over that of k-NN (0.28 from Figure 3.2.2).

4.2.3 Species Prediction by Linear Regression – ROC and Recall/Precision

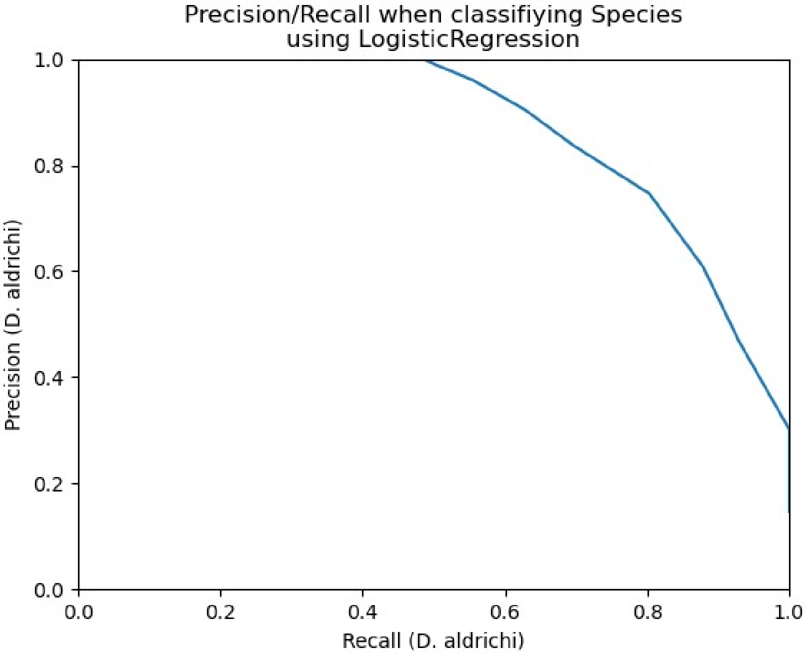
Figure 4.2.3.1



Reading from the graph, to obtain a TPR of 0.9, we must endure a FPR of at least 0.6.
Area Under Curve was calculated as 0.851.
These facts simply reinforce the idea that it is harder to predict species than it is to predict the sex of an individual.

For completeness, the Precision/Recall graph has been determined as well.

Figure 4.2.3.2

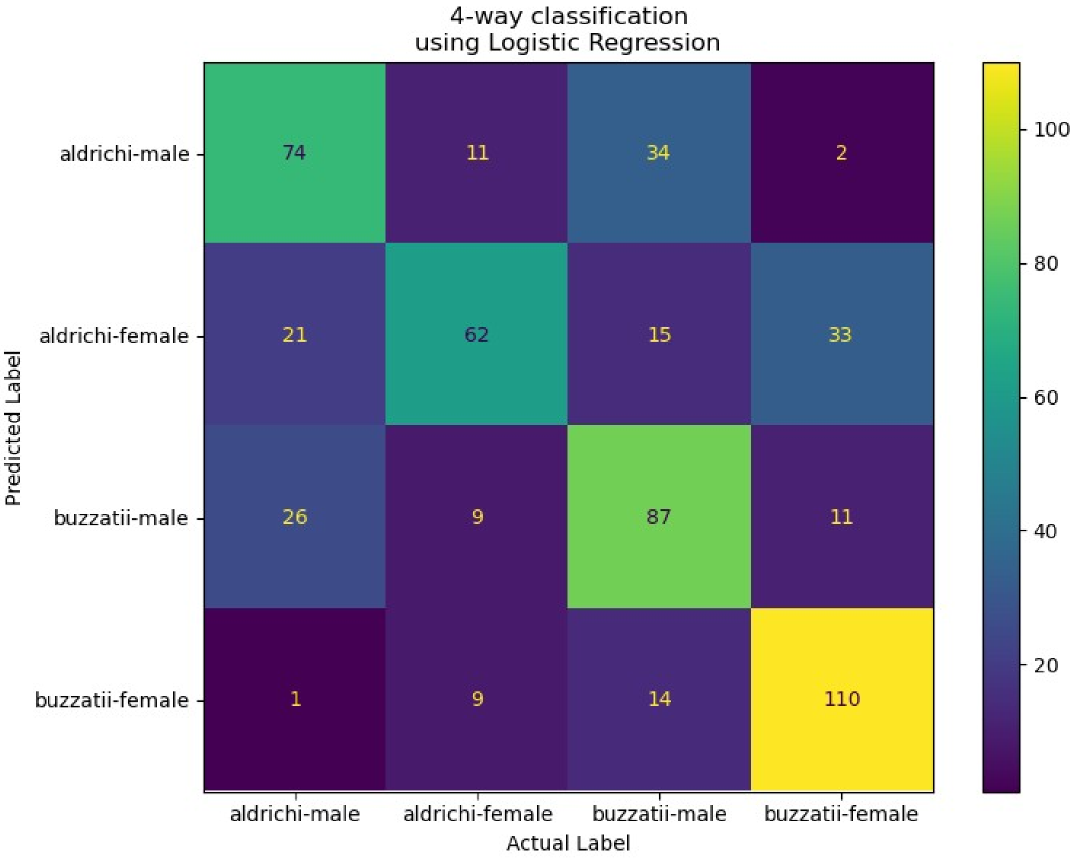


Area Under
Curve: 0.725

4.3 Four-class Classification using Logistic Regression

Linear regression was also used to perform a 4-way classification. Here, the liblinear solver with L1 regularisation. The solver was instructed to minimise misclassification error on a “one versus the rest” basis.

Figure 4.3.1



The overall error rate was 0.358. This is a slight improvement when k-NN was used to for the problem (0.405 from Figure 3.3.1).

If we assume that probability of correctly predicting sex (0.848) as independent of the probability of correctly predicting species (0.788), we can multiply these values together to give 0.668. This corresponds to an error rate of 0.332 which is slightly lower than the observed error rate of 0.358.

The model appears to identify the *D. buzzatii*-female class quite well. If we consider the bottom row of the confusion matrix, 110 out of 134 individuals were correctly identified as *D. buzzatii* females (recall = 0.821). Again, this appears to be at the cost of over predicting this class. If we look at the rightmost column, 156 individuals are predicted to be *D. buzzatii* females of which only 110 were correct (precision = 0.705).

Chapter 5 – Discussion

The CSV (comma separated values) files supplied for this assignment contained raw data for a scientific paper published by Loeschcke *et al*. That paper was a study of fruit flies – *Drosophila aldrichi* and *Drosophila buzzati*. In my report I concentrated on the data supplied in the file 83_Loeschcke_et_al_2000_Thorax_&_wing_traits_lab pops.csv. This file contained data on 1700 specimens and was roughly equally divided between species and between sexes. It contained various physical measurements of each individual such as thorax length.

The objective of this assignment was to employ some of the machine learning techniques on this “real world” data. Two binary classification problems were obvious from the dataset. Firstly, was it possible to build a model which could accurately predict the sex of an individual? Secondly, could we build another model to predict the species? A third classification problem also presented itself: was it possible to perform a 4-way classification combining species and sex?

In any endeavour, it often best to attempt the simplest solutions first and determine if they perform adequately before moving on more complex solutions. This assignment concentrated on the perhaps two of the simplest machine learning techniques: k-Nearest Neighbour and Logistic Regression.

Both of these techniques performed quite well when predicting the sex of a fruit fly. Logistic Regression (with L1 regularisation) had an accuracy of approximately 0.85 while k-NN (k=11) had an accuracy of 0.83. The “feature selection” of L1 regularisation made it clear that 4 features had the most influence on the Linear Regression model. They were thorax length, l2, l3d and w3. An even simpler Linear Regression model, using only thorax length, perform well. It had an accuracy of 0.78. This suggests that thorax length is the major influencing feature when attempting to predict sex. A visual inspection of the raw data indicates that the females seem to be larger than males. It may be that machine learning techniques are overkill for sex classification. Even simpler statistical methods might be tried in the future.

The models predicting species had a slightly lower accuracy: 0.79 for Linear Regression and 0.72 for k-NN (k=11). This may be due to the fact that no one feature dominates the species prediction the way thorax length strongly predicts sex.

The Linear Regression models appear to out-perform the k-NN models for both sex and species. No further analysis has been performed to determine if this is statistically significant.

The models generated for the 4-way classification performed as expected if we assume that the successful prediction of sex and species are independent random variables.

In summary, these simple machine learning models (k-NN and Logistic Regression) were more than adequate for these binary classification problems. K-NN is considered even simpler than Logistic Regression. It has the added advantage that is may be possible to explain k-NN to a client (or supervisor) who does not have a machine learning or mathematical background.

Constraints of time and space have prevented the exploration of more sophisticated techniques such as neural networks.

References

Loeschcke, V., Bundgaard, J. and Barker, J.S.F. (2000) *Heredity* **85**, pp 423-433

Lindholm, A., Wahlstrom, N., Lindsten, F. and Schon, T.B (2022)*, Machine Learning, A First Course for Engineers and Scientists*, Cambridge University Press

List of Abbreviations

AUC	area under curve
CSV	comma separated values
FPR	false positive rate
k-NN	k-Nearest Neighbours
ROC	received operator characteristics
TPR	true positive rate

Appendix A

The following are reproduction of various figures and tables from Linholm *et al*.

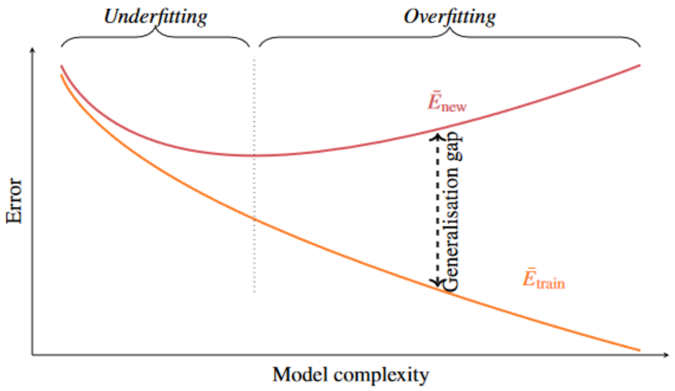


Figure 4.3: Behavior of $\bar{E}_{\text{train}}$ and $\bar{E}_{\text{new}}$ for many supervised machine learning methods, as a function of model complexity. We have not made a formal definition of complexity, but a rough proxy is the number of parameters that are learned from the data. The difference

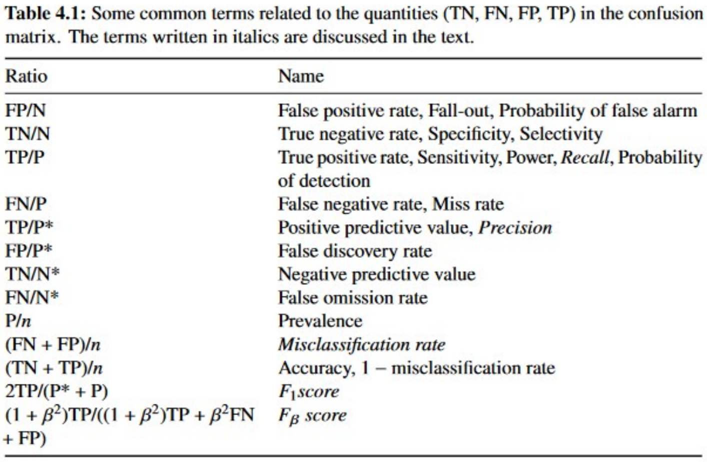
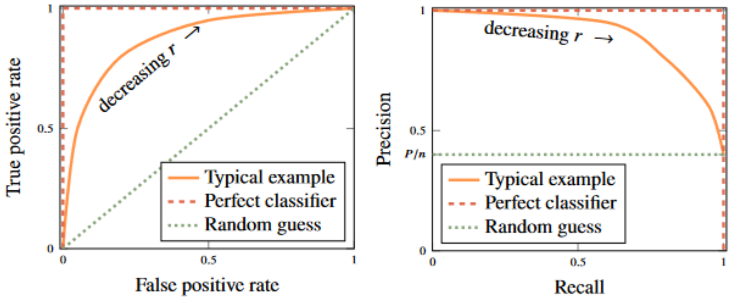


Table 4.1: Some common terms related to the quantities (TN, FN, FP, TP) in the confusion matrix. The terms written in italics are discussed in the text.

Ratio	Name
FP/N	False positive rate, Fall-out, Probability of false alarm
TN/N	True negative rate, Specificity, Selectivity
TP/P	True positive rate, Sensitivity, Power, *Recall*, Probability of detection
FN/P	False negative rate, Miss rate
TP/P*	Positive predictive value, *Precision*
FP/P*	False discovery rate
TN/N*	Negative predictive value
FN/N*	False omission rate
P/n	Prevalence
(FN + FP)/n	*Misclassification rate*
(TN + TP)/n	Accuracy, 1 − misclassification rate
2TP/(P* + P)	F_1*score*
$(1+\beta^2)$TP/($(1+\beta^2)$TP + β^2FN + FP)	F_β *score*



(a) The ROC curve

(b) The precision–recall curve

Figure 4.13: The ROC (left) and the precision–recall (right) curves. Both plots summarise the performance of a classifier for *all* decision thresholds r (see (3.36)), but the ROC curve is most relevant for balanced problems, whereas the precision–recall curve is more informative for imbalanced problems.