

COMP4702 – Report

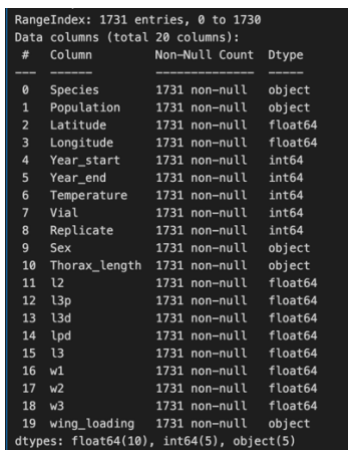
Shanmukh Valluru 46984434

1. Objective:

The task at hand is to demonstrate understanding and the ability to apply relevant machine learning techniques for the given dataset which contains data about *Drosophila* (aka Fruit Flies). The objective will be to find a classification problem which will require the development of a model that has high predictive performance on unseen data. The report will follow through the steps and breakdown individually the implementation of a range of machine learning models and the analysis of the results will be detailed.

2. Finding a Classification Problem:

To find a classification problem, first the dataset needs to be explored to find suitable relevant features and clearly defined classes and make sure it is cleaned and viable for usage on appropriate machine learning algorithms. The dataset which will be used is the '83_Loeschcke_et_al_2000_Thorax_&_wing_traits_lab' csv file.



```
RangeIndex: 1731 entries, 0 to 1730
Data columns (total 20 columns):
#   Column              Non-Null Count  Dtype
---  --
0   Species             1731 non-null   object
1   Population           1731 non-null   object
2   Latitude            1731 non-null   float64
3   Longitude           1731 non-null   float64
4   Year_start          1731 non-null   int64
5   Year_end            1731 non-null   int64
6   Temperature         1731 non-null   int64
7   Vial                1731 non-null   int64
8   Replicate           1731 non-null   int64
9   Sex                 1731 non-null   object
10  Thorax_length       1731 non-null   object
11  l2                  1731 non-null   float64
12  l3p                 1731 non-null   float64
13  l3d                 1731 non-null   float64
14  lpd                 1731 non-null   float64
15  l3                  1731 non-null   float64
16  w1                  1731 non-null   float64
17  w2                  1731 non-null   float64
18  w3                  1731 non-null   float64
19  wing_loading        1731 non-null   object
dtypes: float64(10), int64(5), object(5)
```

Figure 1 - General Summary of the dataset

This dataset contains 20 columns and 1731 rows of collected information from different fruit fly samples (See figure 1). With most of the columns being numerical, the choice of the variable we aim to predict lies between 'Species', 'Population', and 'Sex' variables as they are the main categorical variables, and the choice of the feature variables will be between the numerical variables. The 'Species' variable has 2 unique values, 'Population' has 6 unique values and 'Sex' has 2 unique values in which we can classify the data.

The distribution of the 3 categorical variables will be analysed to see the balance of the distribution of values. As having an even distribution for the target variable is important as it allows for a balanced training which ensures that each class gets adequate representation in the training dataset. As by having an imbalance in distribution this can have a negative impact on the performance of the model as it leads to biases which cause for overfitting for the majority class and will lead to poor generalisation for unseen data.

All three categorical variables have even distribution which means that they are all fit for being allocated as the target variable. As shown by Figure 2 below, the distribution of the 'Species' variable, it can be seen that the distribution is even as the count of 'D._aldrichi' equates to 840 and the count of 'D._buzzatti' is 891.

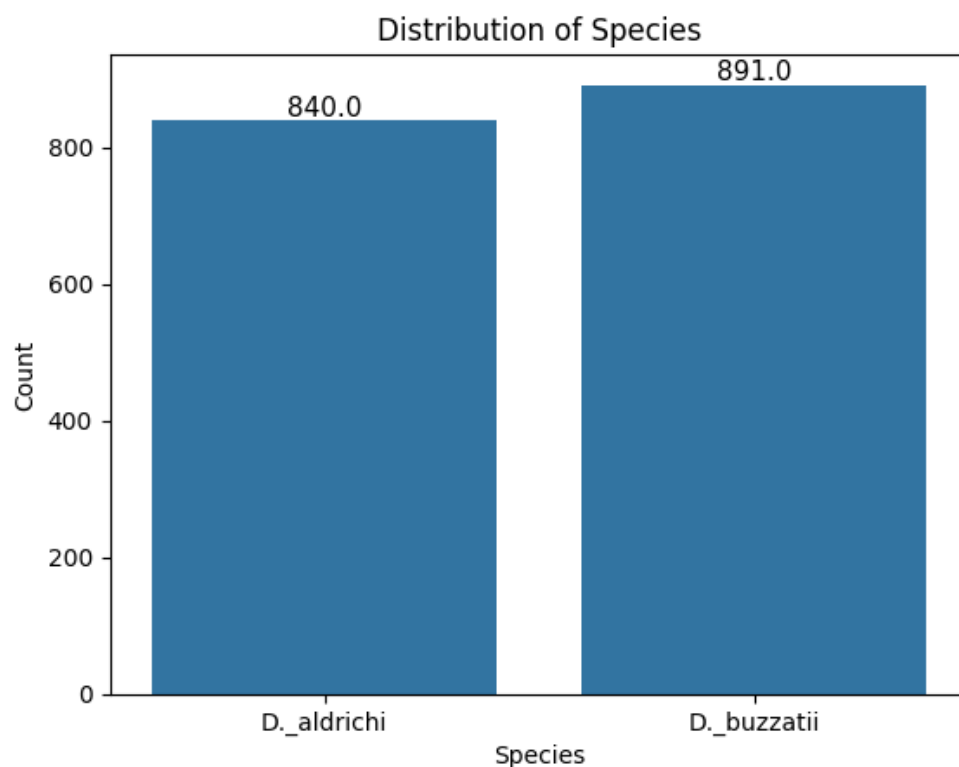


Figure 2 - Distribution of the 'Species' variable

Since 'Species' variable has two unique values, 'D._aldrichi' and 'D._buzzatii' which are the different types of fruit flies, this makes for an intriguing target variable and will lead to a binary classification problem. However, to be able to justify 'Species' as the target variable, further exploration of the relationships between the variables in the dataset is needed.

To get an idea of the correlation and strength of relationships between variables in the dataset a correlation heatmap will be produced. The heatmap will allow to identify potential candidates for the feature variables and clarify which categorical variable can be used as a target variable for the classification problem. To create the heatmap, the non-numerical columns had to be encoded using LabelEncoder from sklearn.preprocessing.

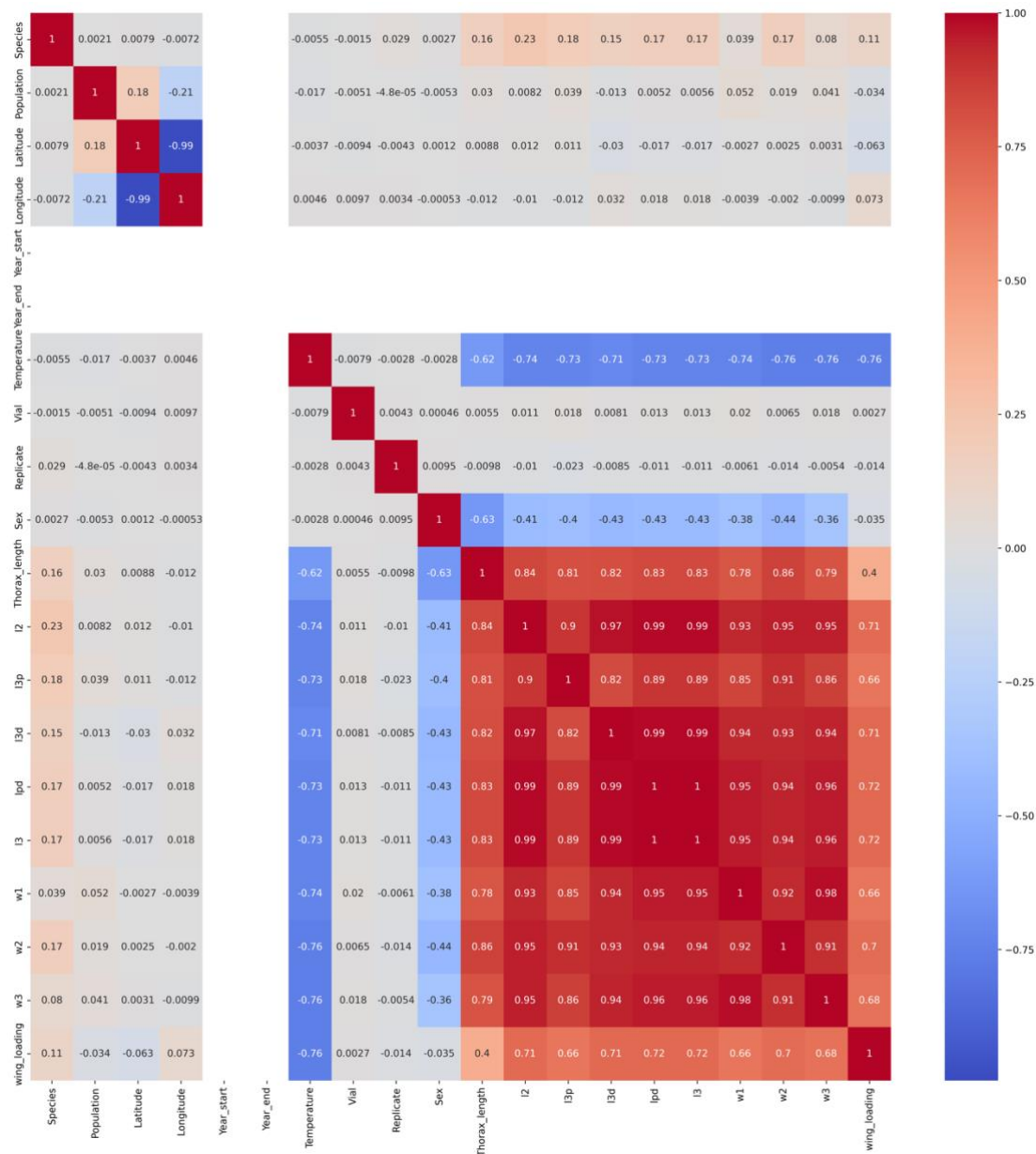


Figure 3 - Correlation Heatmap of the dataset.

Figure 3 above shows the graphical representation of the correlation matrix for the dataset. It shows there is correlation factor between 'Species' and multiple numerical variables such as 'Thorax_length', 'l2', 'l3p', 'l3d', 'lpd', 'l3', 'w1', 'w2', 'w3' and 'wing_loading' which can be assumed as being identified as the measurement features of each fly. This assumption can lead to a reason for the problem as the two species of flies can have different measurements. It can also be observed that there is heavy correlation between the measurement features themselves, which will be explored in more detail in Bivariate EDA section of the report. Since there is correlation between 'Species' and other features, 'Species' will be arbitrarily chosen as the target variable for this classification problem as the heatmap shows that this can be a target variable that can be explored further.

To choose the appropriate classification algorithm from the range of machine learning classification algorithms available, this will be a choice based on the characteristics of the

dataset and the chosen problem. The process of feature selection to identify which of the dataset's features will be effective for machine learning use and predicting the target variable will be conducted more in depth. But first data pre-processing and exploratory data analysis (EDA) will be conducted to make sure the dataset is clean and get a better understanding of the dataset to establish the relationships between the variables and identify the relevant features to predict the 'Species' target variable and complete the classification problem.

3. Data pre-processing:

In this step the dataset will be thoroughly analysed for data quality issues and therefore will be cleaned for machine learning use. Note that this is an iterative process and during the EDA process if further data quality issues are identified the method of handling that issue will be clearly reported.

3.1 Data cleaning:

First the dataset needs to be checked if there are any missing values to ensure a complete dataset and allow for machine learning algorithm use.

```
Species      0
Population   0
Latitude     0
Longitude    0
Year_start   0
Year_end     0
Temperature  0
Vial         0
Replicate    0
Sex          0
Thorax_length 0
l2           0
l3p          0
l3d          0
lpd          0
l3           0
w1           0
w2           0
w3           0
wing_loading 0
dtype: int64
```

Figure 4 - Count of missing values for each column

Figure 3 above shows that there are no missing values for all columns in the dataset. An issue that had to be handled was that even though 'Thorax_length' and 'wing_loading' were numerical features, they were listed as type object. For example, if the value was 7.5 it was recorded as '7.5' which makes it an object. This can be an issue as this can impact the analysis that can be conducted for the dataset. To address this issue, the below code was used to manipulate the data type from object type to numeric type.

```
numeric_data = df[["Thorax_length", "l2", "l3p",
                  "l3d", "lpd", "l3", "w1", "w2", "w3", "wing_loading"]]

numeric_data = numeric_data.apply(pd.to_numeric, errors='coerce')
```

However, this resulted in a single missing value in both 'Thorax_length' and 'wing_loading' columns. This may be due to an error when trying to convert the non-numeric values to numeric values. To deal with this, the row where the missing values are located will be deleted as there are only 1 missing value in both columns.

```
numeric_data = numeric_data.dropna(subset=['wing_loading', 'Thorax_length'])
```

After the above line of code was implemented, it was found that both missing values for 'Thorax_length' and 'wing_loading' referred to the same row. Therefore, only one row was deleted as a result which in return does not affect the loss of information too much depending on the distribution of the columns which will be explored further. It also does not introduce bias into the dataset.

3.2 Univariate Exploratory Data Analysis (EDA):

In this section, each numerical column will be explored in isolation to identify distribution and/or any existing data quality issues. Each column will be described and analysed to give an idea as to which columns will be suitable for being used as feature variables.

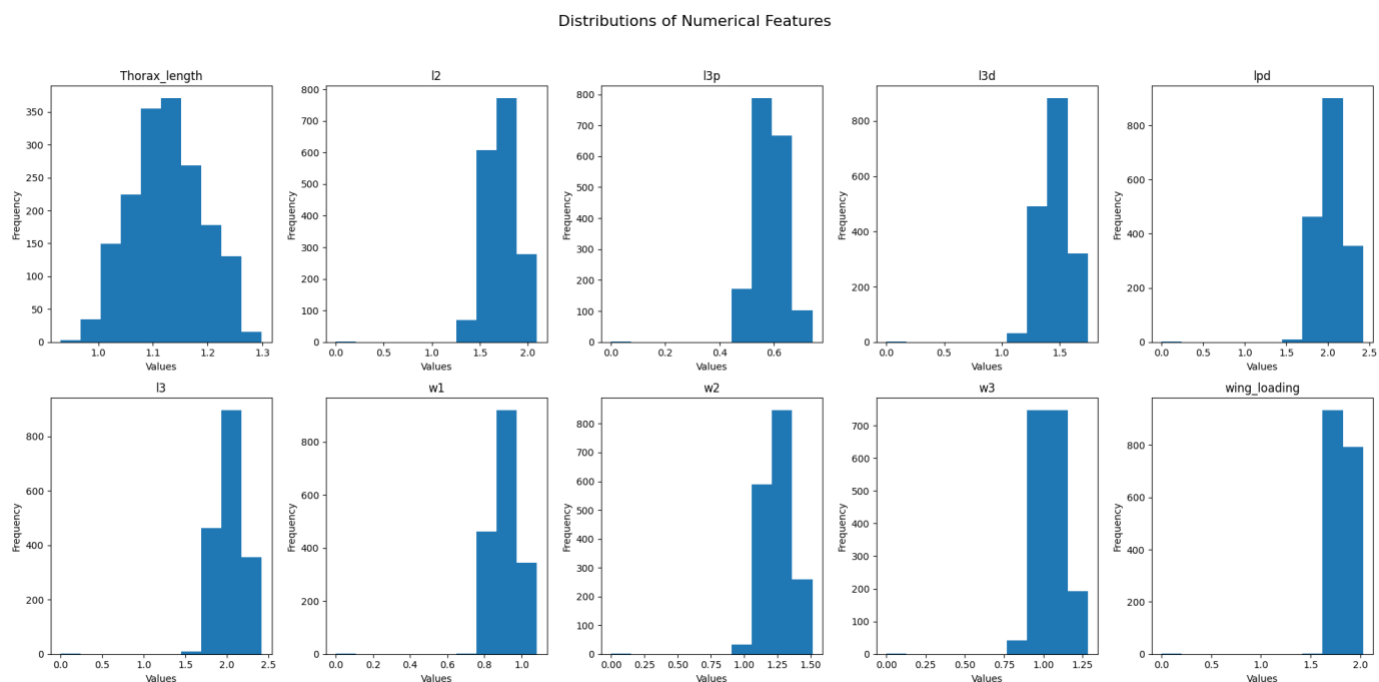


Figure 5 - Distributions of all the numerical columns in the dataset.

It can be observed from Figure 5 that most of the distributions for the columns are normal, however there is presence of outliers. Having outliers can affect model accuracy and its ability to generalise to unseen data, therefore it is important to handle the outliers to improve model performance. A boxplot for each variable will be produced to visualise the impact of the outliers on the distribution more clearly.

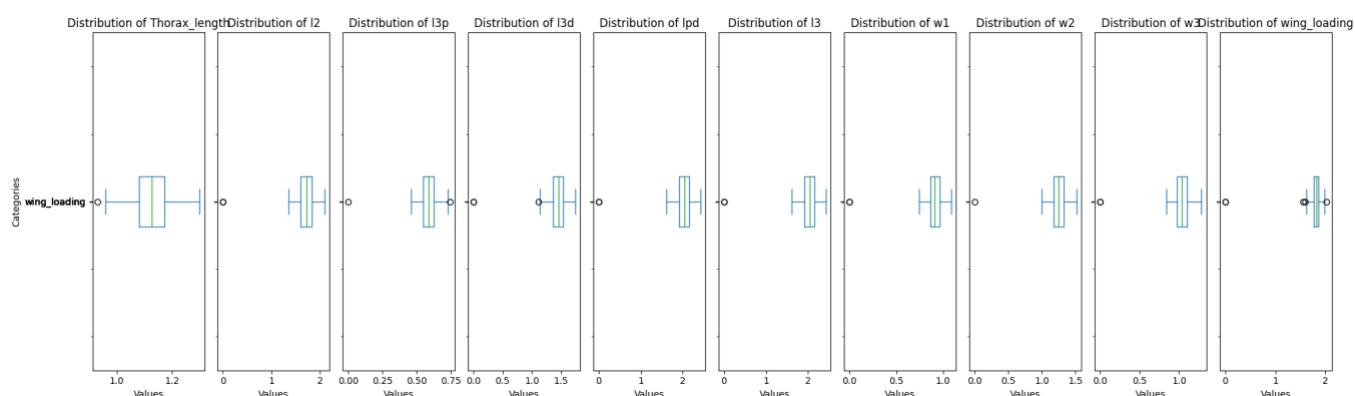


Figure 6 - Boxplot of distribution for numerical features.

Figure 6 shows that most of the outliers that are affecting the distribution of the variables apart from 'Thorax_length' is coming from 0 values which can be assumed it may be from a measurement or data entry error.

Variable	'Thorax_length'	'l2'	'l3p'	'l3d'	'lpd'	'l3'	'w1'	'w2'	'w3'	'wing_loading'
# of 0 values	0	2	1	2	2	2	2	1	2	2

Table 1 - Table of number of 0 values for each numerical feature.

Table 1 shows the number of 0 values for each numerical variable. The choice to remove the rows that contain 0 values will be made on the basis that it will improve the model performance. Note that by removing the rows that contain 0 values which are considered outliers, it will only remove 2 rows from the entire dataset which in return results in now 1728 rows.

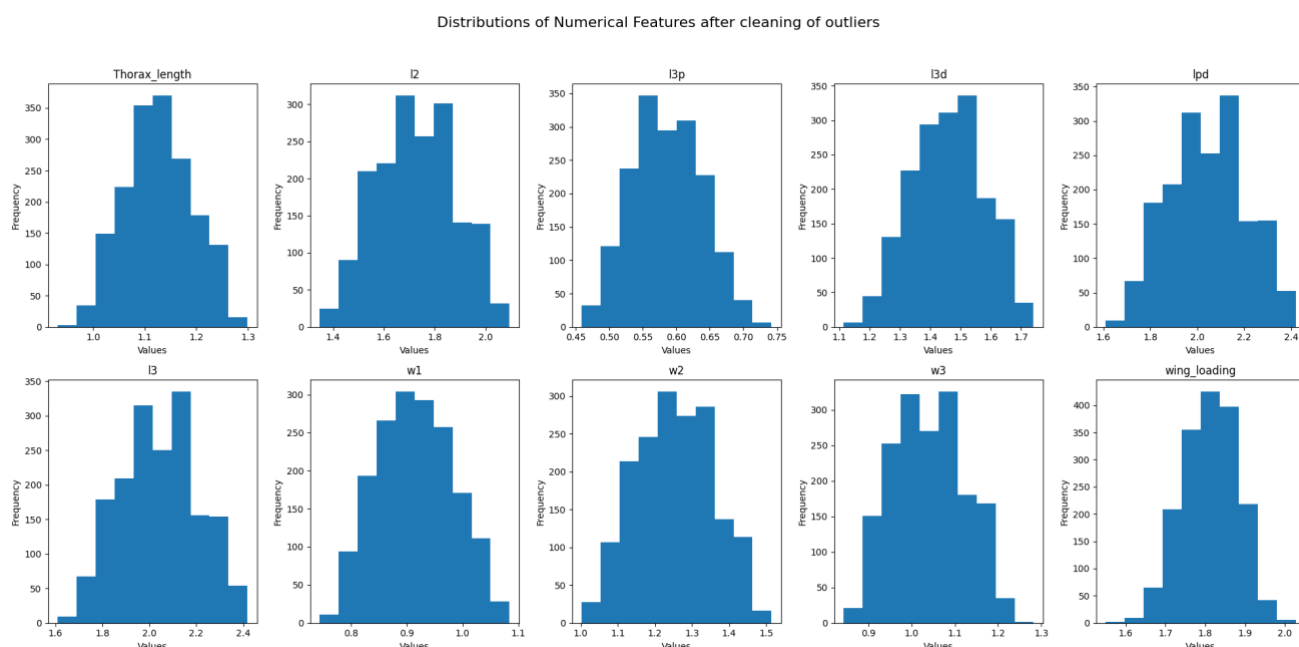


Figure 7 - Distribution of numerical features after cleaning of outliers (0 values).

It can be visualised the difference in distributions of the features before and after removing the 0 value outliers in the dataset as they are less skewed. The consideration of the benefits of removing outliers outweigh the impact of loss of data as now the model will not be trained on outliers which will impact the generalisation and performance of the model.

3.3 Bivariate Exploratory Data Analysis (EDA):

With the process of bivariate EDA, we will be looking at relationships between columns to identify possible trends and patterns to provide relevant/interesting information. This process will enable us to decide the features which will be used for the classification problem.

Firstly, as identified in Figure 3, the heat map showed that the correlation between the numeric variables which are assumed to be measurement values for the fly in the dataset was high. To explore this further, a draftsman display will be produced with just the numerical features.

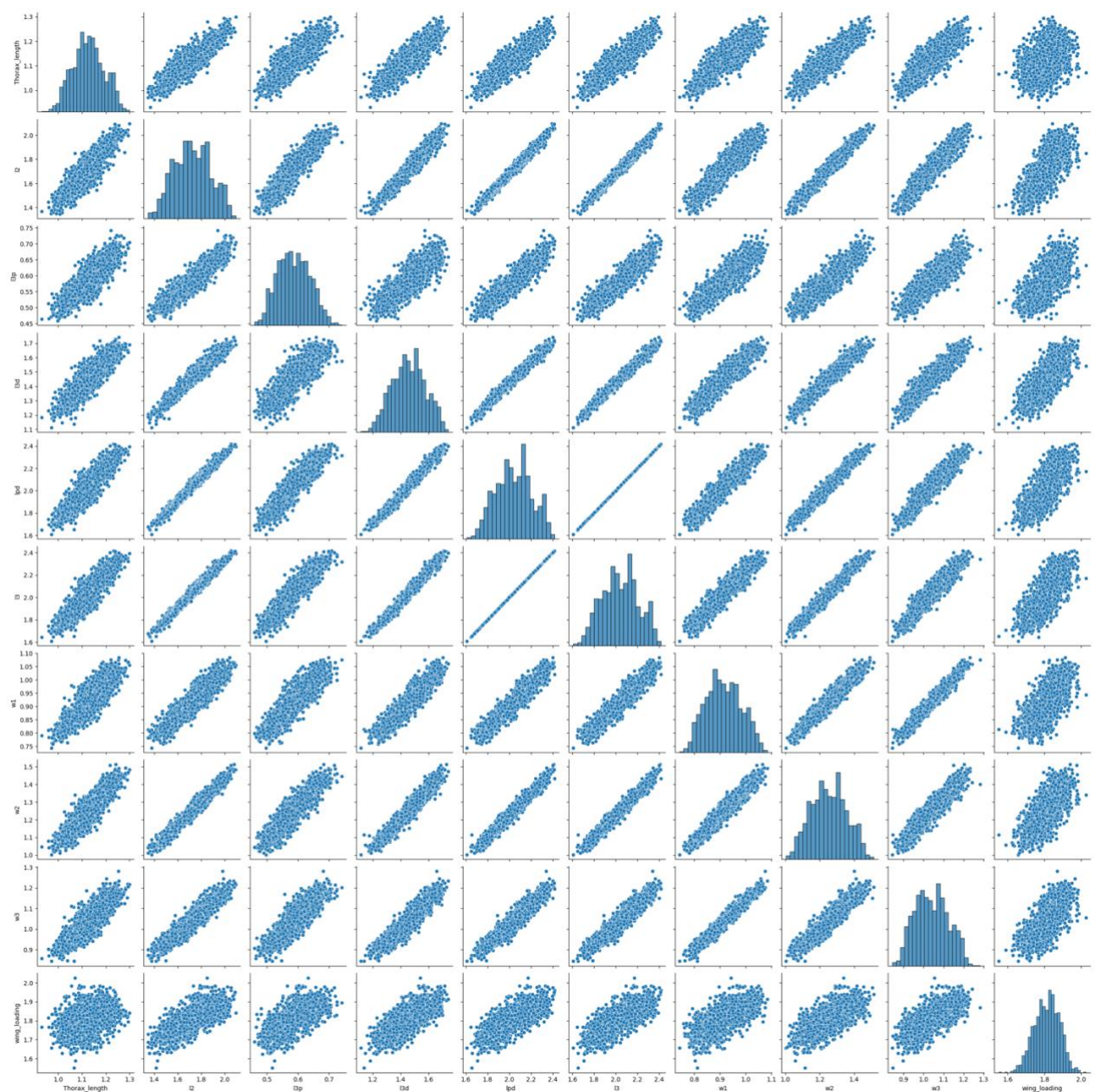


Figure 8 - Draftsman display of feature variables.

Figure 8 shows the correlations between each numeric feature, where there is positive correlation and displays strong linear relationship between many of the features. However, 'wing_loading' can be identified as having the least amount of correlation within the features as its scatterplots regarding other features are more widespread. Notably, the correlation values for 'lpd' and 'l3' seem almost identical as from Figure 7 the correlation coefficient is equal to 0.99. To explore that further a scatter plot and a histogram will be produced between the 'lpd' and 'l3' variables to visualise the correlation and distribution more closely.

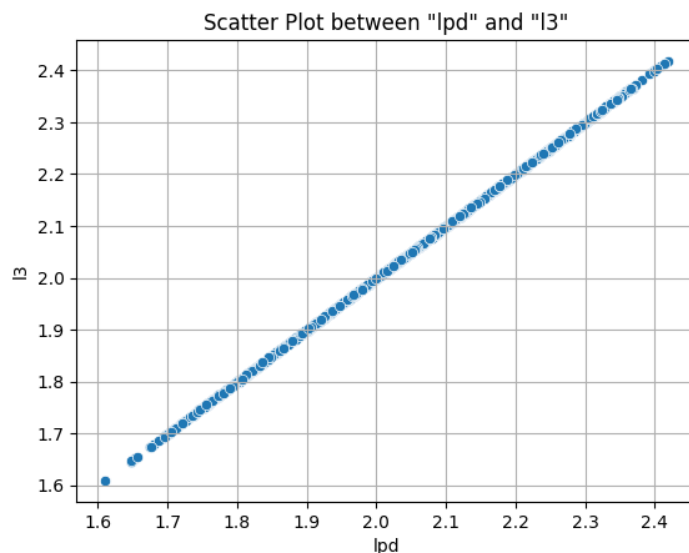


Figure 9 - Scatterplot for 'l3' vs 'l3'

The above figure shows the scatterplot between 'l3' and 'l3' and it can be observed that all the datapoints fit onto a near perfect diagonal line, which supports that the correlation coefficient was found being almost 1 with 0.99.

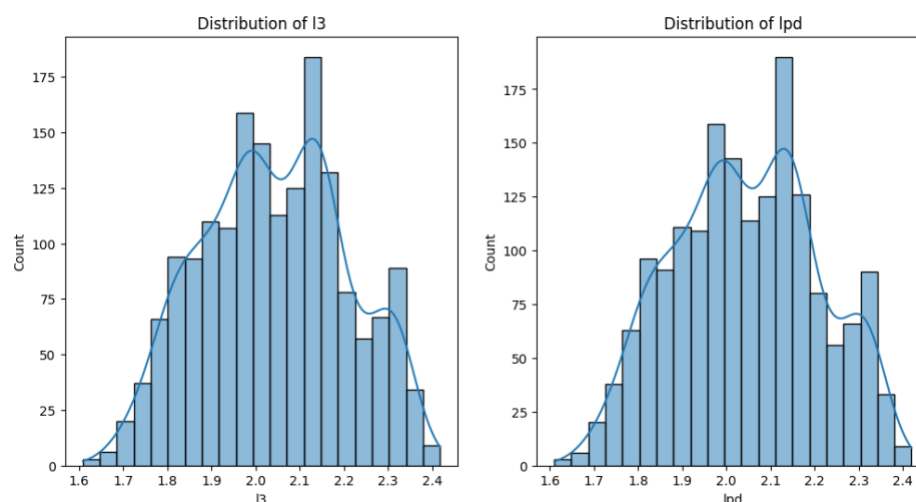


Figure 10 - Comparison of distribution plots for 'l3' and 'l3'.

Figure 10 shows that the distribution of 'l3' and 'l3' are almost identical. Since the scatterplot being a near diagonal line, the distributions of both variables being almost identical and the correlation coefficient equalling 0.99 it can be justified to remove one of the feature variables from the dataset as it can be identified as being redundant. As this would

allow to reduce multicollinearity which affects the interpretability and stability of classification models which means that having highly correlated values, it can lead to misleading relationships between the feature and target variable. The decision to remove one of the either 'l3' or 'lpd' can be supported by choosing to keep the feature which has a higher correlation with the target variable ('Species').

```
correlation1 = numeric_data[['Species', 'lpd']].corr().iloc[0, 1]
print(f'Correlation coefficient between Species and lpd: {correlation1:.5f}')
correlation2 = numeric_data[['Species', 'l3']].corr().iloc[0, 1]
print(f'Correlation coefficient between Species and l3: {correlation2:.5f}')
```

✓ 0.0s

```
Correlation coefficient between Species and lpd: 0.16689
Correlation coefficient between Species and l3: 0.16754
```

Figure 11 - Code and result for correlation coefficient for 'l3' and 'lpd'.

Since the correlation coefficient of 'l3' is higher than 'lpd' as it has a value of 0.16754 in comparison with 0.16689. The reason for choosing the variable with a higher correlation coefficient as it helps with the model performance as it contains more valuable information for predicting the target variable.

Now that the 'lpd' column has been removed, the remaining numeric variables will be used as the features for the classification problem as they all have correlation with 'Species' target variable, which indicates that they may provide useful information for training the model to make accurate predictions. Therefore, the classification problem will be being able to classify data into either 'D._alridchi' or 'D._buzzatii' which are the two distinct classes for the 'Species' target variable (Binary classification problem). This will be done by using the feature variables 'Thorax_length', 'l2', 'l3p', 'l3d', 'w1', 'w2', 'w3' and 'wing_loading'.

```
Index: 1728 entries, 0 to 1730
Data columns (total 10 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   Thorax_length    1728 non-null   float64
1   l2               1728 non-null   float64
2   l3p              1728 non-null   float64
3   l3d              1728 non-null   float64
4   l3               1728 non-null   float64
5   w1               1728 non-null   float64
6   w2               1728 non-null   float64
7   w3               1728 non-null   float64
8   wing_loading     1728 non-null   float64
9   Species          1728 non-null   int64
dtypes: float64(9), int64(1)
```

Figure 12 - Summary of data frame that will be used for machine learning use.

Figure 12 shows the summary of the dataset after the data pre-processing stage, making it suitable for machine learning application. A range of machine learning techniques will be applied to this dataset and the results and outputs will be thoroughly explored and analysed. As a baseline model, the simple yet intuitive k-Nearest Neighbour classifier will be used for the binary classification problem.

4. k-Nearest Neighbour Classifier:

A k-Nearest Neighbour (k-NN) classification model for a binary classification problem such as this works by examining the nearest neighbours up to a specified value of k for each data point. The closeness of these neighbours is determined in an n -dimensional space, identifying the nearest points based on the smallest distance, then the data point classified based on the majority class amongst its k nearest neighbours.

4.1 Training & Evaluation method

To train the model and fine tune the hyperparameter k value, we will use 10-fold cross validation method after using 80/20 training test split for the entire dataset. This approach utilises the training dataset for both the training and evaluation phase instead of using a regular hold-out test set as this ensures that the model is assessed on multiple subsets of data. This provides a more accurate estimation of the model's performance, as the training dataset will be divided into 10 folds (equal parts), and the process of training and validating is performed 10 times. This means each data point in the training set gets a chance to be in validation set exactly once which ensures model gets evaluated on all training dataset which can help reduce bias and can help with avoiding overfitting/underfitting. By getting both cross-validation accuracy from the training set and the test accuracy from the model's performance on completely unseen data, this will ensure a thorough assessment of the k-NN performance.

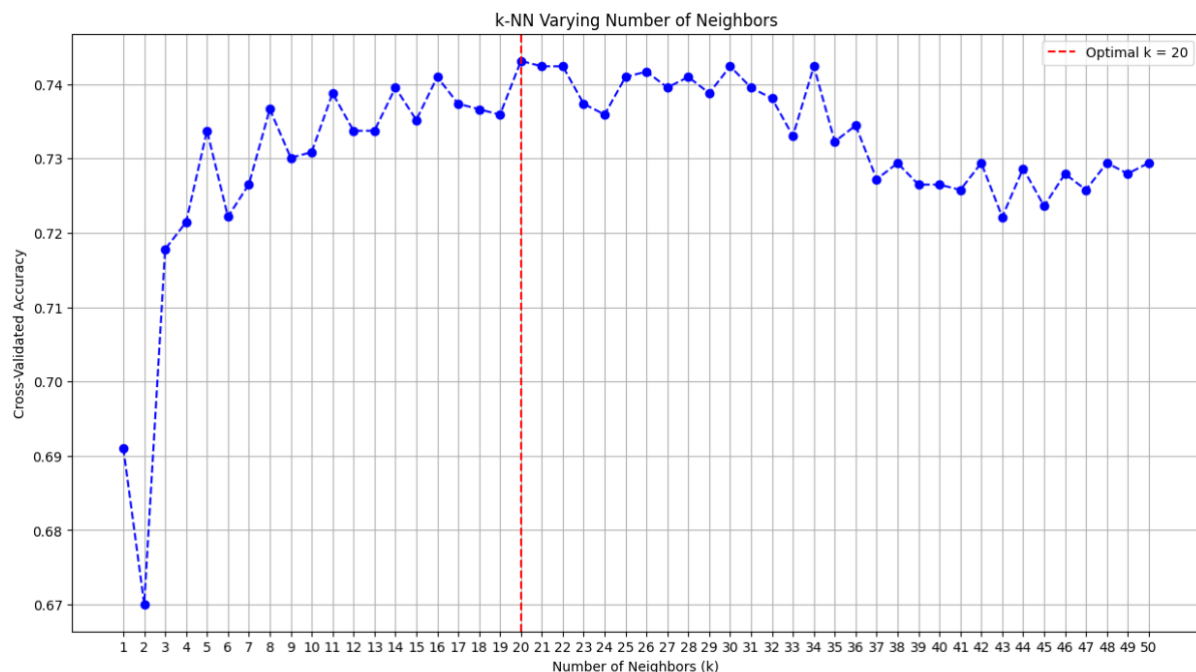


Figure 13 - Cross validation accuracy for a different k -values.

From Figure 13, we can see the cross-validation accuracies for different k values for the model. From $k = 1$ and $k = 4$, the accuracy improves significantly but stabilises after $k = 4$. The plot shows that the optimal k value (red line) where the model has the best cross-validation accuracy is where $k = 20$ with a cross-validation accuracy of 0.7431. Therefore, a k-NN model will be trained using this k value and will be evaluated based off the test set.

4.2 Results & Analysis

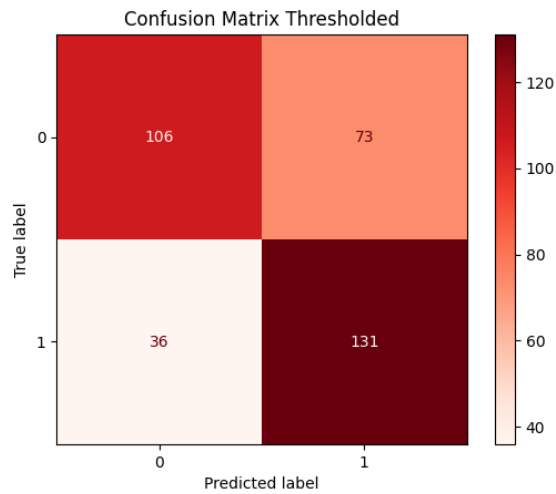


Figure 14 - Confusion matrix to evaluate model.

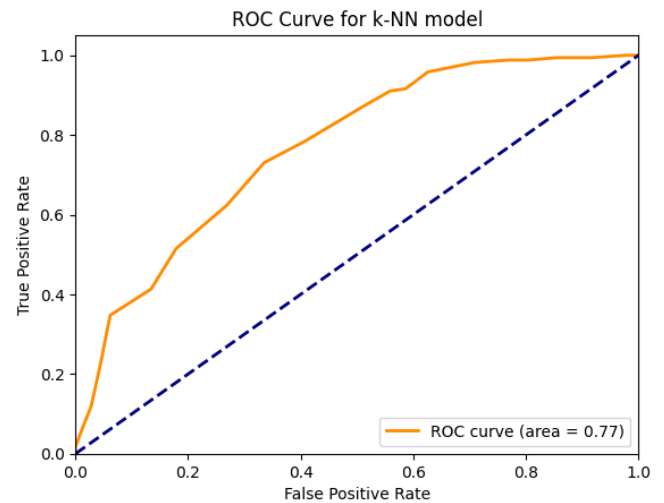


Figure 15 – ROC curve to evaluate model.

The k-NN classifier with k value equal to 20 resulted in test accuracy of 0.6850 when evaluated on the test set. Figures 14 and 15 show a confusion matrix and a ROC curve which are used to evaluate the performance of the model. The confusion matrix shows that the model correctly classified 131 instances of 'D._buzzatii' (represented as 1 on the matrix) flies and 106 instances of 'D._aldrichi' (represented as 0 on the matrix) flies. However, it also classified 73 instances of 'D._aldrichi' and 36 instances of 'D._buzzatii' incorrectly. Therefore, it suggests that the model can be biased to classify data points towards 'D._buzzatii'. This could be due to the 'D._buzzatii' having a higher distribution as demonstrated by Figure 2. Figure 15 shows that the area under the curve is equal to 0.77 which indicates the model has demonstrated capability of discriminating between the classes as a score closer to 1 is perfect, however a score of 0.5 suggests that the model is randomly guessing.

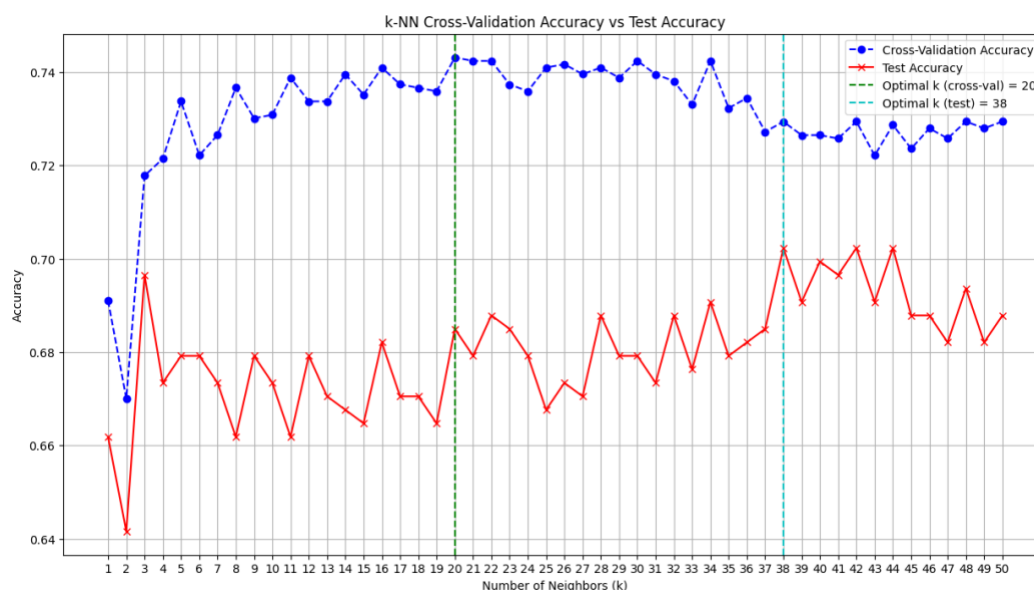


Figure 16 - Cross-validation and Test accuracies for different k values.

By plotting the cross-validation and test accuracies of different k values on the same plot we can compare how well the model performs with respect to training cross-validated subsets and the test set. Figure 16 shows that the model best performs with k = 20 on the training subsets with a cross-validation accuracy of 0.743 and performs best with k = 38 with the unseen test with accuracy of 0.702. It is evident that the model performs marginally better with the training set than the test set as with k=38 it has a lower cross-validation accuracy with 0.729. Since with k = 38 has the higher test accuracy, it can be said that the model will generalise better to unseen data.

Therefore, with the consideration of both cross-validation accuracy of the training subsets and test accuracy of the unseen test set when tuning the hyperparameter k value, it allowed us to determine that the model with k =38 would be the better choice due to its ability to generalise with unseen data. Overall, the k-NN classifier provides for a solid baseline model, however, there is potential room for improvement for accuracy of predicting the 'Species' variable by exploring more sophisticated models. Such classification model is the logistic regression model, which will be the next machine learning technique implemented for the dataset to compare its performance with the k-NN classifier.

5. Logistic Regression Classifier:

A logistic regression model will be used for the dataset to compare its performance with the k-NN classifier model. In section 4 we trained a k-NN model and evaluated it based on the unseen test set, based on the optimal k value it gave a test accuracy of 0.702. However, by intuitively presuming that a more complex model would yield a greater test accuracy, we will analyse the performance of a logistic regression model on the dataset. The logistic regression model provides a more complex approach as it involves the process of modelling a discrete outcome (binary) given input variables. It increases the model complexity by estimating the coefficients for each feature in the dataset which requires an optimisation problem. In our case, the logistic regression model utilizes the sigmoid function given by:

$$f(x) = \frac{1}{1 + e^{-k(x-x_0)}}$$

Equation 1: Sigmoid function

and the input variables and maps between value of 0 or 1 ('D._aldrichi' or 'D._buzzatii') which is the given probability.

5.1 Training & Evaluation method

For the logistic regression model, the training of the model will involve 10-cross validation with the original 80/20 train test split like we performed with k-NN to keep it fair so the comparison between the models will be more viable. The model will then be evaluated by

generating the test accuracy with the model's performance in regard with the test set and will be visualised using a confusion matrix. To try optimising the model to be able to correctly classify the data points into either 'D._aldrichi' or 'D._buzzatii' we will try varying the decision threshold as the default threshold value for a logistic regression model is 0.5. The threshold value is critical for classification as it determines the class membership. For example, if the predicted probability is greater than 0.5 then it will be classified as 1 ('D._buzzatii') and if below 0.5 it will be classified as 0 ('D._aldrichi').

However, as shown by Figure 2, there is a marginal imbalance of distribution between the two 'Species' classes which could impact the sensitivity of the classification and affect the model performance. To explore this, the impact of different threshold values on the model's accuracy will be visualised by plotting accuracy against different threshold values. This will be an important step to which will allow us to select the optimal threshold value and help improve the model performance of the logistic regression model.

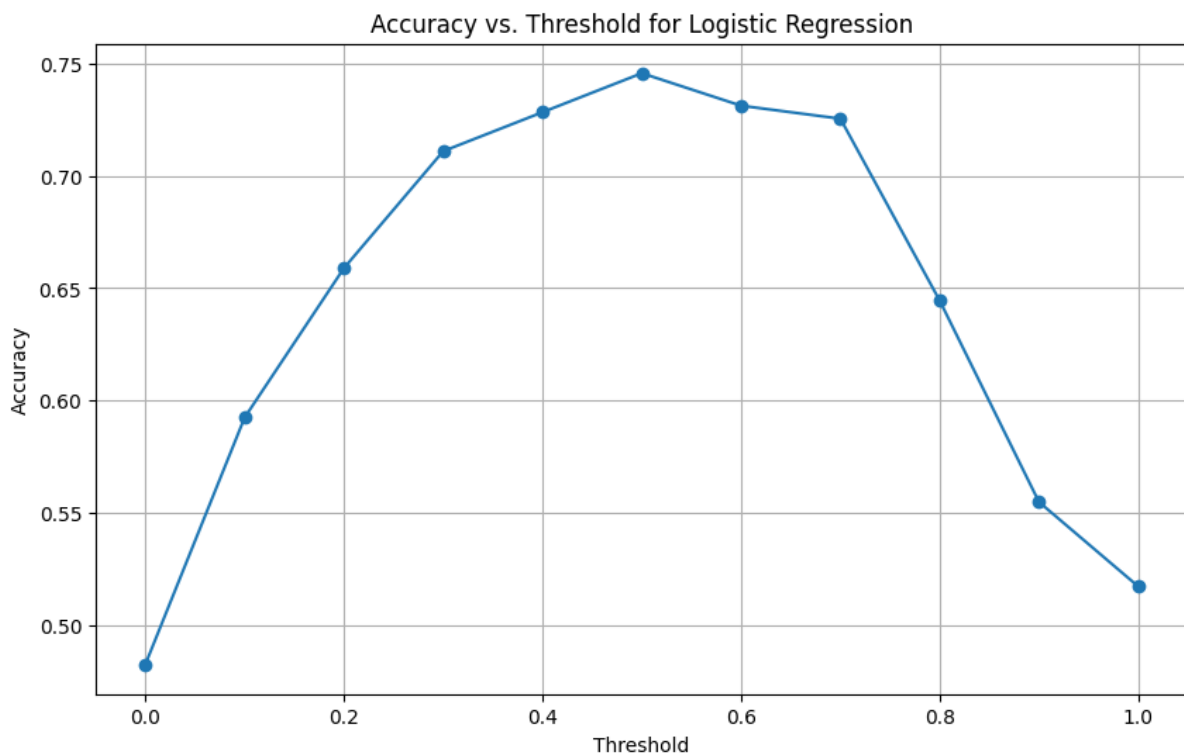


Figure 17 - Plot of accuracy w.r.t different threshold values.

Figure 17 shows how the accuracy of the model changes with respect to the various threshold values. We can see that the accuracy follows a negative quadratic curve shape where it peaks around the midpoint which is approximately around 0.5. Therefore, as the plot suggests 0.5 which is the default threshold for classification, is the optimal threshold.

5.2 Results & Analysis

The logistic regression model achieved a mean cross-validation accuracy of 0.771 and standard deviation of the cross-validation accuracy of 0.032 which suggests that the model is

relatively consistent across the different folds. Now the model will be evaluated based on the unseen test set to analyse the test accuracy and visualise the performance of the model through a confusion matrix and ROC curve.

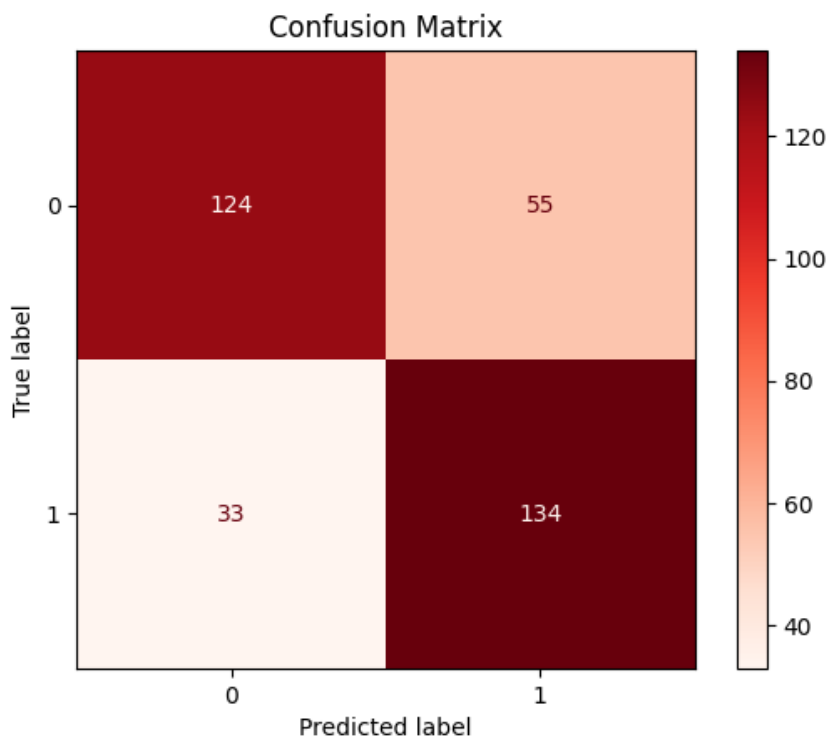


Figure 18 - Confusion matrix for logistic regression function.

Figure 18 shows that the number of correctly predicted 'D._buzzatii' flies (represented as 1) equate to 134 and the number of correctly predicted 'D._aldrichi' (represented as 0) equates to 124. However, it also classified 33 'D._buzzatii' and 55 'D._aldrichi' incorrectly.

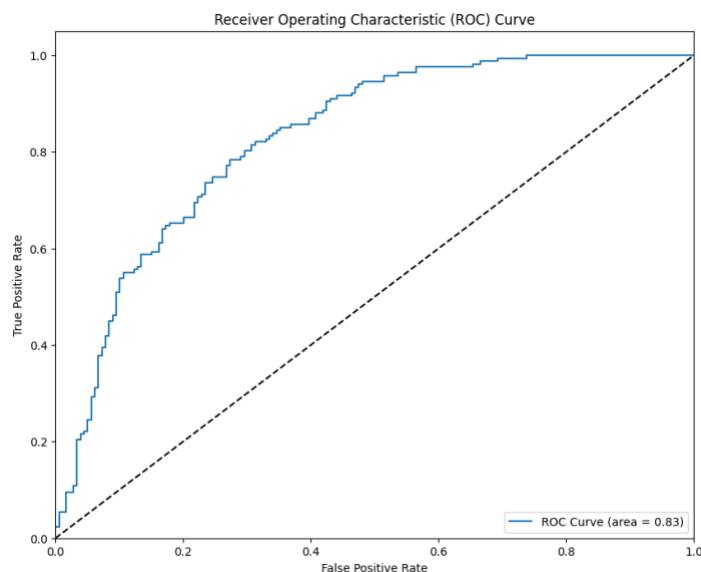


Figure 19 - ROC curve for Logistic Regression model.

The above plot demonstrates the ROC curve for the logistic regression model. It shows that ROC curve always stays above the diagonal line which represents the performance of a

random classifier, this means that the model performs better than random. The area under the curve is equal to 0.83 which means that the model can distinguish between the two 'Species' with 83% probability. Overall, the logistic regression model overall had a test accuracy of 0.746. To analyse the model further, a visualisation will be shown of the coefficient of the feature variables which will provide information for the relationships between each feature variable and the log-odds of the target variable.

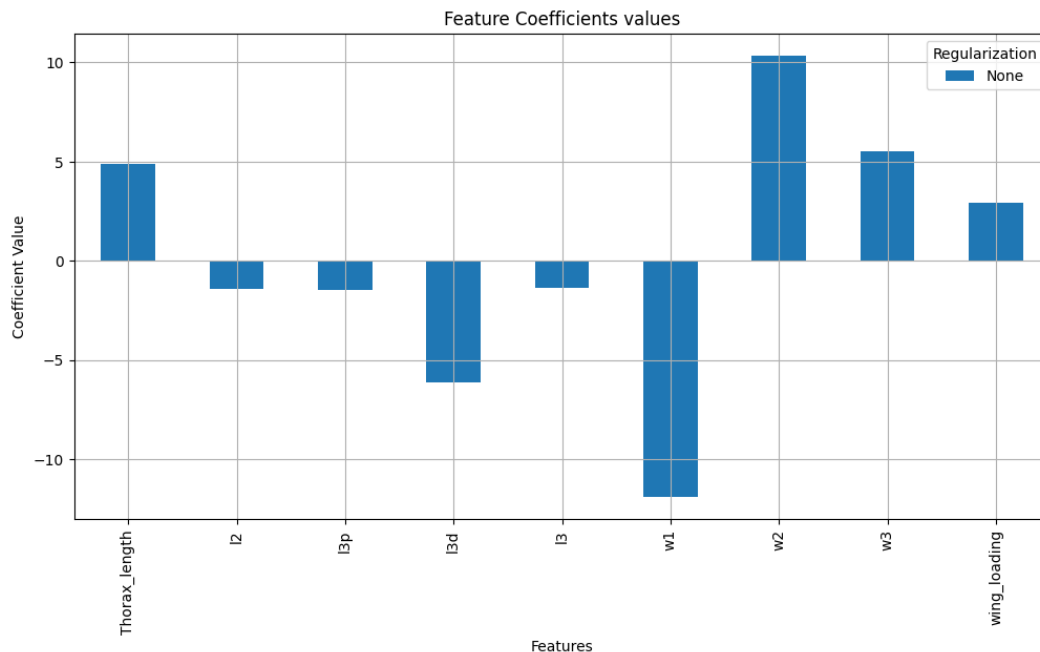


Figure 20 - Coefficient values for Logistic Regression model.

Figure 20 shows the influence of each feature variable on being able to predict the 'Species' variable. 'Thorax_length', 'w2', 'w3' and 'wing_loading' have positive coefficient values with 'w2' having the greatest which means these features have the greatest influence. However, 'l2', 'l3p', 'l3d' and 'l3' all have negative coefficient values which means they have a negative influence on the target variable. This could be due to having high multicollinearity between the feature variables which the dataset has by looking at Figure 8 this can impact the coefficient values as they can be distorted.

In comparison with the baseline k-NN model, the logistic regression model performed better regarding being able to predict the 'Species' of a fly based on feature variables. The logistic regression model resulted in having a better cross-validation and test accuracy than the k-NN classifier as the cross-validation of logistic regression was 0.771 in comparison with k-NN with 0.7431. With the comparison of the test accuracies, logistic regression and k-NN both had a drop from cross-validation accuracy to test accuracy, which was to be expected, however the k-NN model had a larger drop which suggests it does not generalise as well as the logistic regression model to unseen data.

There are many reasons why the logistic regression model performed better than the k-NN model. The logistic regression model could have benefitted from it being a binary classification task, there being strong linear relationships throughout the dataset and could also be because k-NN models are sensitive to noisy data whereas logistic regression models can generalise better. The logistic regression model can also be improved further by adding

an additional layer of complexity by involving regularisation, which will be explored more in the next section.

6. Logistic Regression Classifier with Regularisation:

Regularisation is a technique used for machine learning models to reduce overfitting by increasing the model's capability to generalise. The increased generalisation comes at the expense of the training error, as the training error is bound to increase. It involves during the training phase, adding an additional penalty to the loss function. It also handles multicollinearity, which can be found in our dataset (refer to Figure 8). From section 5, we can see that the logistic regression model had a cross-validation accuracy of 0.771 and test accuracy of 0.746. To try close the gap between the cross-validation accuracy and the test accuracy we will train, test, and compare the performance of both L1 and L2 regularisation methods.

6.1 Training & Evaluation method

Cross-validation will be performed on each regularisation with a range of regularisation strengths (C-value). The C-value is a hyperparameter which controls the strength of the regularisation and needs to be tuned during the process to ensure we get the optimal C-value for each regularisation.

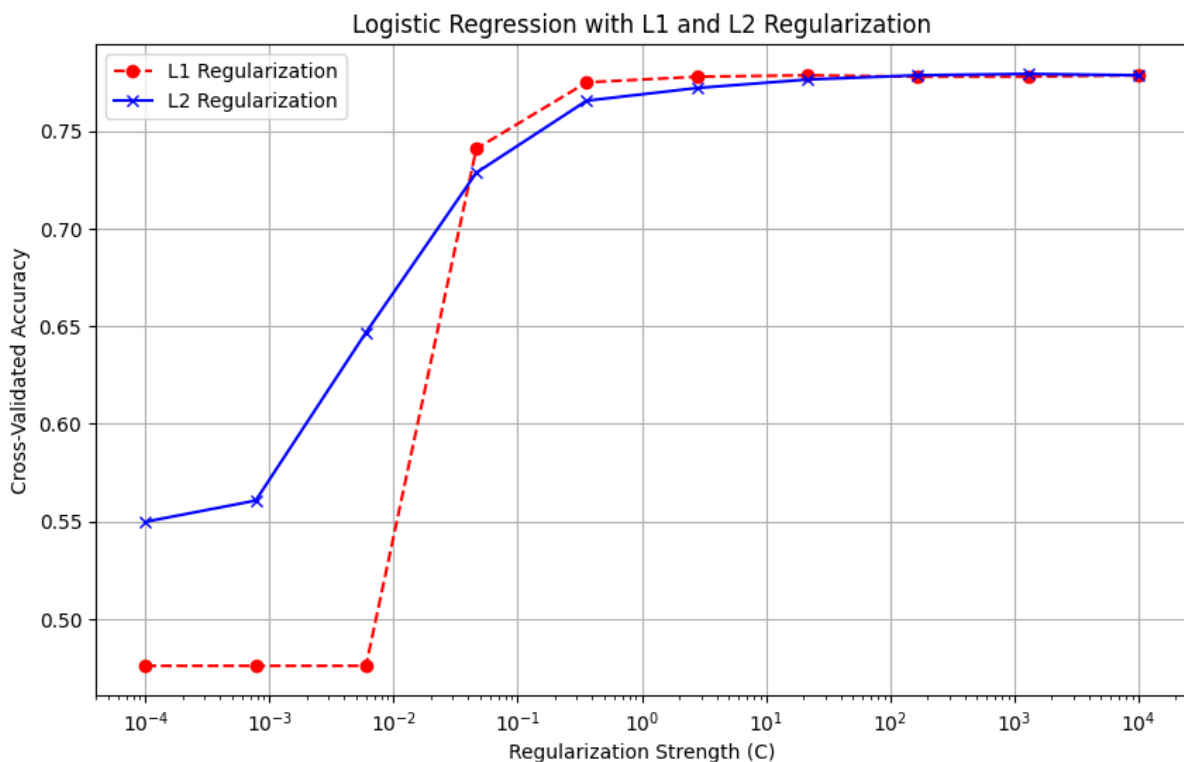


Figure 21 -Comparison of cross-validation accuracy and regularisation strength for Logistic Regression models with L1 and L2 Regularization.

Figure 21 shows that the both L1 and L2 converge to the same cross-validated accuracy of 0.779 as the C value increases. However, the test accuracy for L1 regularisation was better as it scored 0.754 with an optimal C-value of 21.544, whereas the L2 regularisation test

accuracy was marginally less with 0.751 with an optimal C-value of 1291.55. Using the optimal C-value, a logistic regression model will be trained and tested using L1 regularisation.

6.2 Results & Analysis

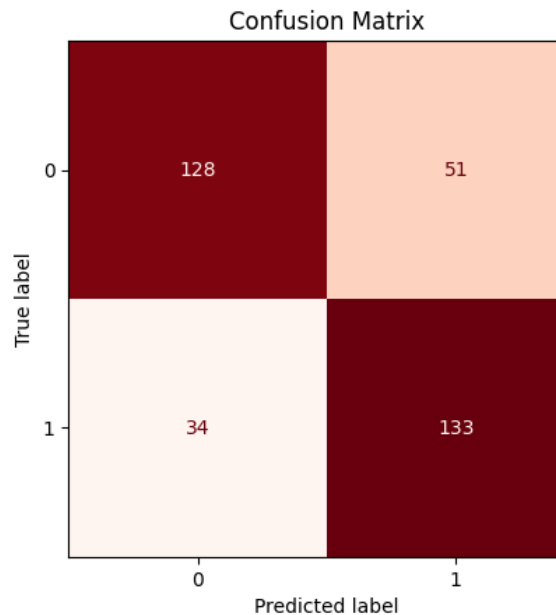


Figure 22 - Confusion matrix for Logistic Regression model with L1 regularisation and optimised C-value.

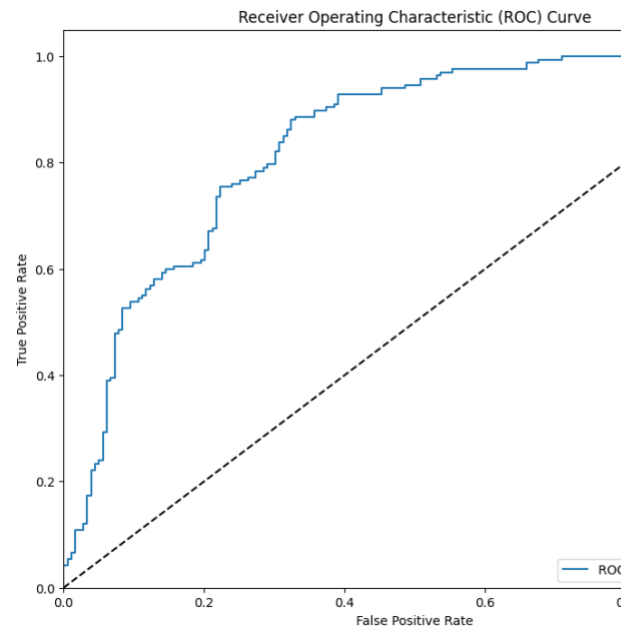


Figure 23 - ROC curve for Logistic Regression model with L1 regularisation.

Figure 22 shows the improved confusion matrix with the added L1 regularisation for the logistic regression model. An interesting observation is that by making the changes to the model, it was able to predict more 'D._aldrichi' flies correctly.

The ROC curve for the model shows an increase in the area under the curve, as it now equals to 0.84. From the confusion matrix and the ROC curve, it can be justified to say that by adding the L1 regularisation with an optimised C-value, it generated a model which was able to predict the 'Species' with accuracy of 0.754. Now we will look at how the feature coefficients of the logistic regression models compare when they have L1, L2 regularisation or none.

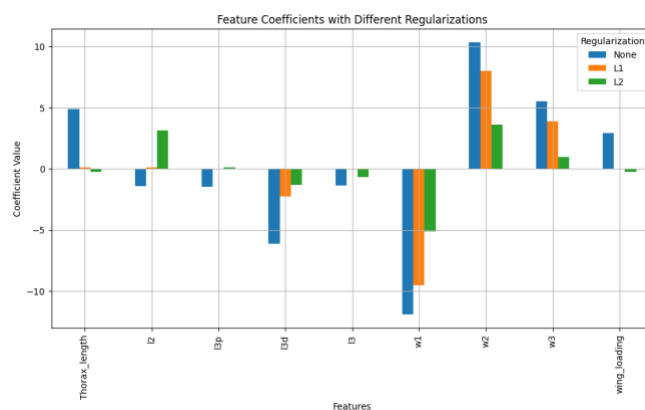


Figure 24 - Feature coefficient values for different regularisations.

Figure 24 shows the direct impact regularisation has on the logistic regression models. L1 regularisation it drives some coefficients to 0 as it effectively performs feature selection by only including the feature variables that are valuable. L2 regularisation decreases the coefficients but does not set to 0 as by doing this it will have evenly distributed coefficients. For our dataset and classification problem, the L1 regularisation was the most effective as it had the highest accuracy on unseen data. Figure 24 shows how L1 regularisation (orange) drives the coefficients of 'Thorax_length', 'l2', 'l3p', 'l3' and 'wing_loading' to either 0 or very minimal. With feature variables 'w1', 'l3d', 'w2' and 'w3' having the larger coefficients. It can be supported by Figure 24 that the features like 'wing_loading', 'l2' and 'l3' have minimal impact in predictive power. Therefore, it can be determined that those features are either redundant or unrelated to the classification problem.

7. Conclusion:

In conclusion, when comparing the different machine learning models being used for the same classification problem of being able to predict the 'Species' given input feature variables. After using k-Nearest Neighbour, logistic regression, and logistic regression with L1 regularisation, we found that logistic regression with L1 regularisation to be the model with the highest predictive performance on the unseen test set.

The following table will show the summary of results of the different models with optimised hyperparameters.

	k-NN	Logistic regression	Logistic regression w/ L1 regularisation
Cross-validation accuracy	0.729	0.771	0.779
Test accuracy	0.702	0.746	0.754

Table 2 – Summary of results.

Table 2 shows how the cross-validation and test accuracy of the last model was highest. More than just finding which model performs the best, we were able to analyse and understand why and how certain components contribute to a model's performance.