Machine Learning
Harsha Joshi
45561623

Fruit-Fly Analysis:
Classifying Thorax Length by Sex

Define Task

The paper by Loeschcke et al (2000) is primarily about wing spans and thorax length of fruit flies found in Eastern Australia. The data collected from the report demonstrates these topics and is organised in the following csv files:

- 83_Loeschcke_et_al_2000_Thorax_&_wing_traits_lab pops: is a general overview of the fruit flies' thorax and wing attributes. There are various distances on the wing, represented by the following attributes: l2, l3p, l3d, lpd, l3, w1, w2, w3.
- 84_Loeschcke_et_al_2000_Wing_traits_&_asymmetry_lab pops: is much more specific in it's focus. This file explicitly compares wing attributes (wing_area, wing_shape, wing_vein) to its asymmetric wing counterparts (asymmetry_wing_area, asymmetry_wing_shape, asymmetry_wing_vein). There is a clear delineation between these two types of attributes.
- 85_Loeschcke_et_al_2000_Wing_asymmetry_lab_pops: continues to increase in specificity. This dataset only exclusively follows the asymmetric measurements of the attributes first introduced in 83_Loeschcke_et_al_2000_Thorax_&_wing_traits_lab pops. As such, 85 is much more clear and condensed in demonstrating how the asymmetric measurements can be used to predict values.

These CSV files will be referred to as CSV file 83, 84 and 85 respectively from this point forward.

Prior to performing an exploratory data analysis, a brief observation of what each of the spreadsheets' attributes cover will be mentioned. Further, and most importantly, the number of null values in all three spreadsheets will be found to determine whether the data presented is viable for modelling. It is important that to create the best machine learning model, the csv file with the most accurate data and least number of empty values is utilised such that important extractions and calculations can be made to pre-existing records to determine a predictive model for future inputs (Durgapal, 2023).

Briefly, it is seen that CSV file 83 contains a large variety of attributes including population, sex, latitude, longitude as parameters of where the data was collected. Similarly, specific to fruit flies their thorax length and certain specific wing attributes (l2, l3p, l3d, lpd, l3, w1, w2, w3) are mentioned in this dataset. If this dataset was to be chosen, a machine learning model could be made based on comparing the attributes specific to fruit flies against a parameter such as sex, population, latitude, or longitude to determine whether gender or location effects the fruit fly specific attributes. Similarly, this dataset has no null values and is a holistic view of fruit flies in their environment.

CSV file 84 is slightly different. It is immediately obvious that there are two categories of fruit fly data both related to the wings split into normal and asymmetric measurements: wing_area, wing_shape, wing_vein versus asymmetry_wing_area, asymmetry_wing_shape, asymmetry_wing_vein. Choosing this dataset in theory could be a great way to compare the differences between normal wing data against asymmetric wing data. However, it was evident

that these attributes all had null values (Table 1). As such, it would not be ideal to construct a machine learning model on a dataset which includes empty points, as this does not provide an accurate distribution in these fields.

CSV file 85 has a similar problem as 84. While the environmental descriptors of where the fruit flies were found contain no null values, the key wing attributes – in this case asymmetry distances – contained null values in each category (Table 1). While 85 is the most specific of the datasets, due to its empty values located on the most important attributes, it may not be viable to construct a machine learning model on this particular dataset.

TABLE 1: *Number of Missing Values in Each CSV File*

CSV File	83	84	85
	('Missing Values:\n' 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	('Missing Values:\n', Species 0 Population 0 Latitude 0 Longitude 0 Year_start 0 Year_end 0 Temperature 0 Vial 0 Replicate 0 Sex 0 Wing_area 1 Wing_shape 19 Wing_vein 6 Asymmetry_wing_area 26 Asymmetry_wing_shape 26 Asymmetry_wing_vein 14	('Missing Values:\n', Species 0 Population 0 Latitude 0 Longitude 0 Year_start 0 Year_end 0 Temperature 0 Vial 0 Replicate 0 Sex 0 Asymmetry_l2 6 Asymmetry_l3p 1 Asymmetry_l3d 9 Asymmetry_lpd 10 Asymmetry_l3 10 Asymmetry_w1 15 Asymmetry_w2 12 Asymmetry_w3 14

Given the brief overview of all three datasets and the fact that two of the three CSV files contain empty values, it is evident that the best action is moving forward is to utilising file 83 to generate a machine learning model. This can be done with either classification or regression.

In 83's case, because it provides an overview of the entire dataset, it would be beneficial to follow a classification machine learning model. This is because attributes can be much more easily compared against one another to demonstrate the impact they have on the dataset as a whole. This can allow us to make better predictions for future values using the constructed machine learning model.

Regression would not be a good fit for 83 as it is defined as "supervised machine learning technique which is used to predict continuous values" (Kurama, 2023). While wing length measurements are continuous values, it would not be beneficial to use a continuous machine learning technique to provide a general overview. In general, which will be proven further in the

data analysis section, it will be seen that the dataset falls under a certain range despite wing length measurements being classified as continuous values.

Data Analysis

To ensure a thorough data analysis, it is evident that first an "Exploratory Data Analysis" (EDA) must be conducted such that conclusions can be drawn, and further analysis can be done to find the most effective classification model.

Firstly, the total number of rows and columns is calculated, and it is evident that there are 1731 rows and 20 columns. Hence, this remains to be a relatively small dataset and can be used in its entirety to train a classification model. A small dataset is defined as being 'small' enough for human comprehension (Wikipedia, 2024). Given that there are only about 1731 rows of data, with about 20 attributes, this dataset is not substantially large such that it is beyond human comprehension.

Further, in printing out the head of the dataset (the first 5 rows), the structure of the data can be seen:

FIGURE 1: *First five rows of 83_Loeschcke_et_al_2000_Thorax_&_wing_traits_lab pops.*

	Species	Population	Latitude	Longitude	Year_start	Year_end	\
0	D._aldrichi	Binjour	-25.52	151.45	1994	1994	
1	D._aldrichi	Binjour	-25.52	151.45	1994	1994	
2	D._aldrichi	Binjour	-25.52	151.45	1994	1994	
3	D._aldrichi	Binjour	-25.52	151.45	1994	1994	
4	D._aldrichi	Binjour	-25.52	151.45	1994	1994	

	Temperature	Vial	Replicate	Sex	Thorax_length	l2	l3p	l3d	\
0	20	1	1	female	1.238	2.017	0.659	1.711	
1	20	1	1	male	1.113	1.811	0.609	1.539	
2	20	1	2	female	1.215	1.985	0.648	1.671	
3	20	1	2	male	1.123	1.713	0.596	1.495	
4	20	2	1	female	1.218	1.938	0.641	1.658	

	lpd	l3	w1	w2	w3	wing_loading
0	2.370	2.370	1.032	1.441	1.192	1.914
1	2.148	2.146	0.938	1.299	1.066	1.928
2	2.319	2.319	0.991	1.396	1.142	1.908
3	2.091	2.088	0.958	1.286	1.062	1.860
4	2.298	2.298	1.010	1.418	1.148	1.886

It is evident that the classification model from this structure can focus on comparing the numerical heavy wing length attributes of: thorax_length, l2, l3p, l3d, lpd, l3, w1, w2, w3 against one (or more) of the attributes that are not numerical. This will allow the model to classify whether the lengths of fruit fly wings/thorax are impacted by factors such as species, population, location, temperature or sex.

To construct a classification based on attributes not related to wing length/thorax and further those that are also non-numerical, it is evident that the number of unique classes in these attributes must be found, such that it becomes clear how the wing length/thorax will be

distributed. In this case, only Species, Population and Sex are non-numerical and have certain classes. The number of unique classes for these three attributes are as follows:

FIGURE 2: *Unique values for Species, Population and Sex.*

```
Unique values for Species : ['D._aldrichi' 'D._buzzatii']
Unique values for Population : ['Binjour' 'Gogango_Creek' 'Grandchester' 'Oxford_Downs' 'Wahruna']
Unique values for Sex : ['female' 'male']
```

Hence, it is evident that Species and Sex have two distinct classes by which wing length/thorax data can be classified, while Population has five distinct classes. Contextually, it is known that Species and Population are unique to Eastern Australia where the study by Loeschcke et al was conducted. As such, though a model would be possible for both these realms, a much more effective and global model that researchers around the world could use would be classifying the distinctions between females versus males for wing length/thorax data. This has the benefit of being applicable at a larger scope than constructing a classification model for species and population that might be only relevant to an Eastern Australia context.

From earlier in the report in Table 1, it is evident that 83 does not have any null values and as such there is no preprocessing required – this is validated by Figure 3. This will allow more effective model to be constructed as there are no missing values preventing classification of the data and a better holistic view is evident as the null values do not have to be handled by estimation and/or deletion.

FIGURE 3: *Validating the none of the attributes in 83_Loeschcke_et_al_2000_Thorax_&_wing_traits_lab pops contain NULL values.*

#	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

Further Data Analysis

While the above provided a great overview of the dataset, further data analysis is needed to construct the ultimate classification model. Now that it has been established that Sex is the best attribute to classify the wing lengths/thorax by, summary statistics of these attributes are established as follows:

TABLE 2: *Summary Statistics for Numerical Features*.

Summary Statistics for Numerical Features:

	Latitude	Longitude	Temperature	l2	l3p
count	1731.000000	1731.000000	1731.000000	1731.000000	1731.000000
mean	-24.794910	150.821693	24.982669	1.723935	0.585854
std	1.958099	1.220711	4.076542	0.165536	0.053610
min	-27.680000	148.850000	20.000000	0.000000	0.000000
25%	-25.520000	150.170000	20.000000	1.607000	0.547000
50%	-25.200000	151.170000	25.000000	1.722000	0.585000
75%	-23.770000	151.450000	30.000000	1.840000	0.624000
max	-21.770000	152.450000	30.000000	2.095000	0.742000

	l3d	lpd	l3	w1	w2
count	1731.000000	1731.000000	1731.000000	1731.000000	1731.000000
mean	1.455826	2.041169	2.040291	0.914038	1.252196
std	0.128044	0.178219	0.178354	0.074163	0.106781
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1.370000	1.920500	1.919000	0.864000	1.176000
50%	1.457000	2.040000	2.039000	0.912000	1.251000
75%	1.540000	2.159500	2.158500	0.963000	1.325500
max	1.742000	2.419000	2.418000	1.084000	1.514000

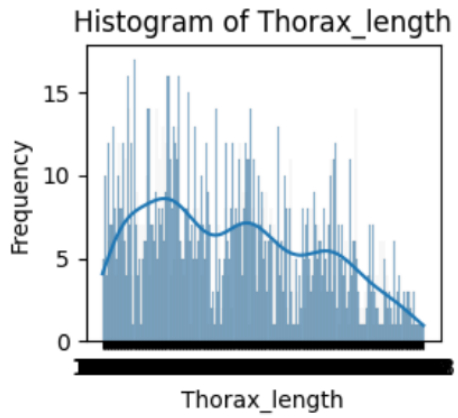
	w3
count	1731.000000
mean	1.038279
std	0.089665
min	0.000000
25%	0.976000
50%	1.037000
75%	1.100000
max	1.282000

Further, a histogram is constructed with these summary statistics to demonstrate the distribution of the data. This allows the establishment of the distribution of data and how wide the classification model must reach to cover the totality of the dataset. Similarly, a histogram can be a visual indication of any outliers or misplaced data if there are unusual visual spikes in the distribution.

Histograms were produced for attributes with numerical data and mapped against frequency to demonstrate their distribution. In this case, latitude, longitude, temperature, thorax length, l2, l3p, l3d, lpd, l3, w1, w2, w3 were mapped.

Out of the produced histograms, the most effectively distributed attribute was thorax length.

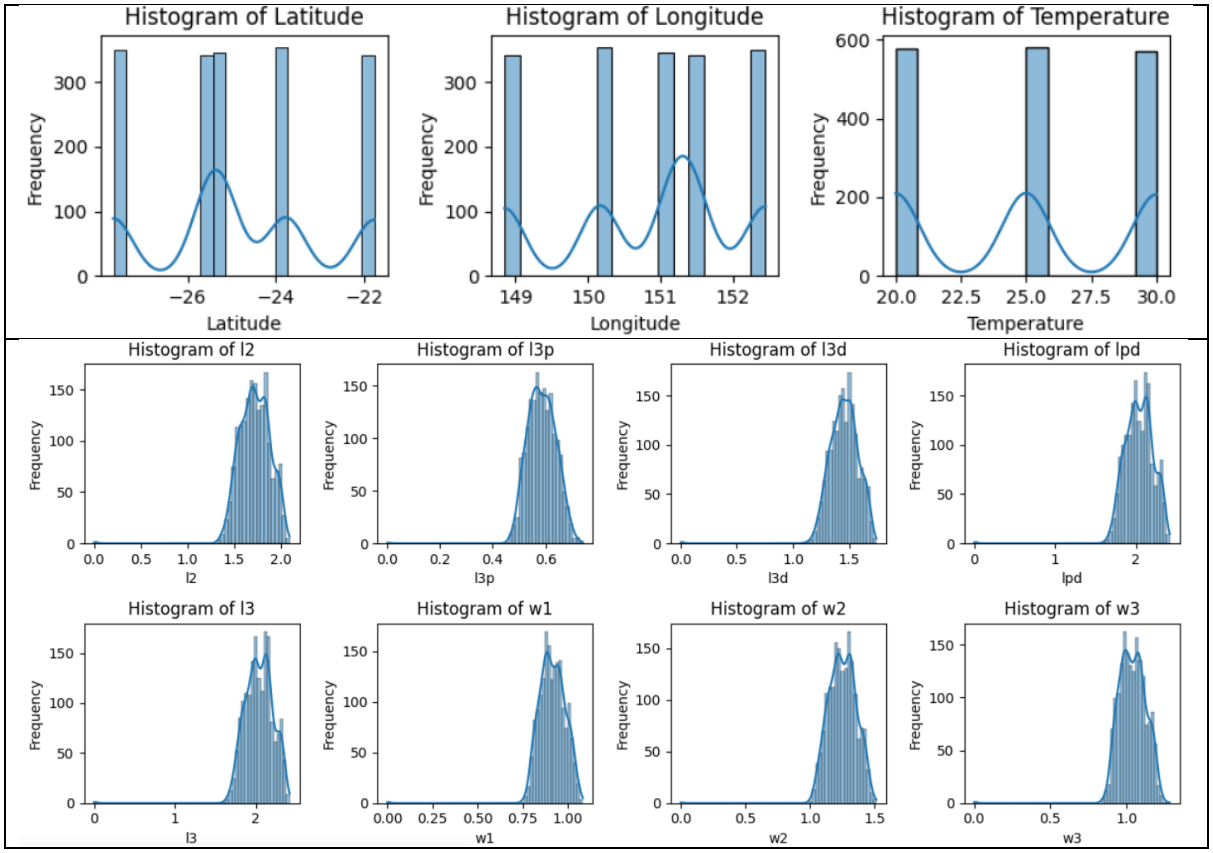
FIGURE 4: *Thorax length histogram*.



Latitude, longitude and temperature all had limited distribution in their histograms as this data was only concerned with a limited type of climate and location, seeing as most of the data was collected in Eastern Queensland. As such these attributes are ineffective to include in a potential classification model under the attribute 'Sex' due to the distribution being so restrictive. Similarly, all the wing length attributes – l2, l3p, l3d, lpd, l3, w1, w2, w3 – had extremely similar distributions. This means that there does not seem to be a distinguishing factor that any of the wing lengths

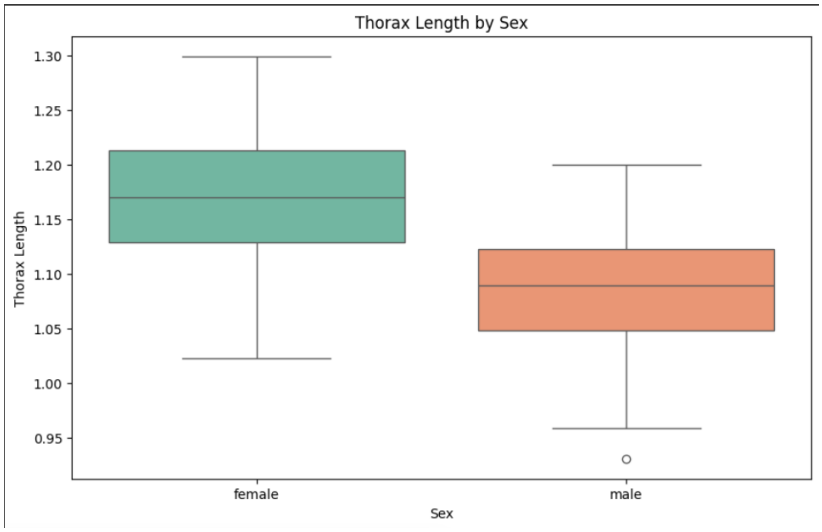
have when it comes to the distribution of their data. As such, any one of the wing lengths would not yield distinct results when classified under 'Sex' either (Table 3).

TABLE 3: *Latitude, Longitude, Temperature, Wing lengths histograms.*



Further, given that the summary statistics in Table 2 also included values related to the interquartile range (25%, 50%, 75%), a boxplot can be constructed. This will be mapping the thorax length by sex attribute only, given that the histograms for the other attributes would not be effective moving forward to create a classification-based model. It is to be noted that the two unique values of the attribute sex will be applied here: male and female. This will allow a clear distinction and classification of the thorax length by gender. The distinction will enable further data analysis and development of the model.

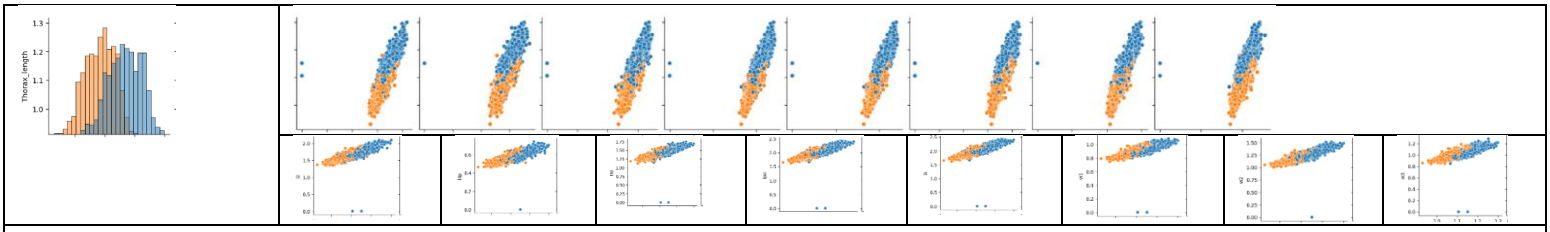
FIGURE 5: *Thorax Length by Sex Boxplot – comparing males versus female distribution.*



Thus, from Figure 5, it is clearly seen that the thorax length of females is on average longer than that of males with a mean of approximately 1.17 against the males' average thorax length of 1.09. Hence, this supports the prediction and data analysis thus far that classifying the thorax length attribute against the sex attribute will yield evident distinguishing factors between the different classes (males versus female) in this case.

Thorax length being the ideal attribute to map against sex, is further evidenced by constructing a pairplot across all numerical attributes against sex as the hue (being the variable in the data to map plot aspects to different colours (Seaborn, 2024). A pairplot, similar to a histogram, reveals patterns, trends and correlations by showing the relationship between two variables (Ahluwalia, 2024). Though the code for this is provided in Appendix 1, focussing on the thorax length against sex attribute demonstrates that there is a clear distinction between the male versus female datapoints. The datapoints for male fruit flies is separate from the female fruit flies very evidently when the graph is visualised:

TABLE 4: *Pairplot of Thorax length against Sex – Males versus Females.*

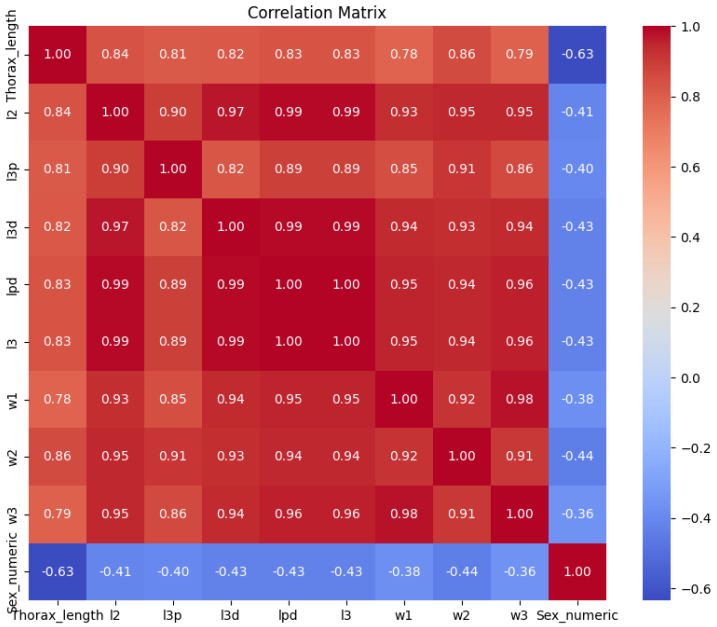


NOTE:

- Legend:
  - females = blue
  - males = orange

The final step in data analysis is to ensure the correlation matrix effectively demonstrates the correlation between thorax length and sex. In the code to construct this, a parameter 'sex_numeric' was constructed from the original 'sex' attribute in the dataset. This is because the original dataset was restricted to 'sex' being male or female. However, the male and female distinct values must be assigned numbers such that the correlation matrix can be constructed for 83. This was done using the label_encoder function which is part of the sklearn.preprocessing library. The correlation matrix is as follows:

FIGURE 6: *Correlation Matrix for Numerical Attributes Against Sex.*

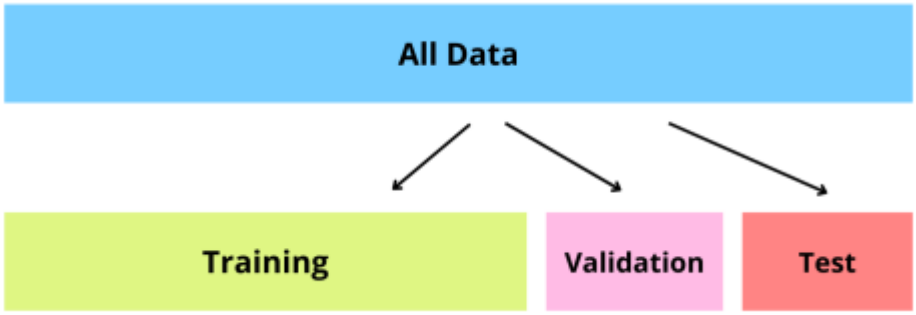


From this it is evident that thorax length has a correlation of 0.63 when compared to sex_numeric's 1.00. It has the highest value of all the numeric attribute's correlations provided. Therefore, thorax length is highly correlated against sex and can effectively be taken further to construct a classification model.

Constructing & Computing the Model

Now that it has been established that thorax length against sex is the most effective numerical attribute to be classified against the sex parameter specifically, further segregation of data is required such that the final model can be effectively tested, trained, and validated.

FIGURE 7: *Train-Validation-Test Data Split* (Chavan, 2023).



As such, some preliminary analysis of how much of the overall model that consists of training and testing data, which will be further utilised to validate the model, is done as follows, with the output printed:

TABLE 5: *Training and Test data – code and output.*

```
from sklearn.model_selection import train_test_split

features = ['Thorax_length', 'l2', 'l3p', 'l3d', 'lpd', 'l3', 'w1', 'w2', 'w3', 'wing_loading']
X = dataset[features]
y = dataset['Sex']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42, stratify = y)
```

```
Training set shape (X_train): (1384, 10)
Training set shape (y_train): (1384,)
Testing set shape (X_test): (346, 10)
Testing set shape (y_test): (346,)
```

It is to be noted that the training and test shape is highly dependent on the parameters set by the test_size and random_state when utilising the train_test_split module from sklearn.model_selection. In this case, it has been specified that approximately 20% (0.2) of the total dataset should be used for testing, as such 346 (20% of 1731) points will be used to test and create the model from. Similarly random_state as 42 is chosen to control the random number generated used to shuffle the data before splitting it. These attributes can be changed and will impact the overall model and its effectiveness. It is important that the entire dataset is not to be used for training nor testing (see that 346 < 1731 and 42 < 1731).

After this various statistical accuracy and classification reports of different types of modelling techniques were utilised to garner which technique would be the most effective to construct a classification model. For this, logistic regression, decision tree, random forest and SVM are all utilised to determine which techniques are most effective towards constructing the final model. Some pseudocode will be provided in Appendix 2 to demonstrate the logic of how the accuracy and classification reports were constructed for each of these five models. Please note, Appendix 2 only cover pseudocode for logistic regression, though the same logic is applicable for the other three modelling techniques.

TABLE 6: *Accuracy and Classification Report Summary of Four Modelling Techniques – logistic regression, decision tree, random forest and SVM.*

Logistic Regression

```
Logistic Regression Accuracy: 0.7947976878612717
Logistic Regression Classification Report:
              precision    recall  f1-score   support

      female       0.82      0.76      0.79       172
        male       0.78      0.83      0.80       174

    accuracy                           0.79       346
   macro avg       0.80      0.79      0.79       346
weighted avg       0.80      0.79      0.79       346
```

Decision Tree	Decision Tree Accuracy: 0.8005780346820809 Decision Tree Classification Report: precision recall f1-score support female 0.81 0.78 0.80 172 male 0.79 0.82 0.81 174 accuracy 0.80 346 macro avg 0.80 0.80 0.80 346 weighted avg 0.80 0.80 0.80 346
Random Forest	Random Forest Accuracy: 0.8323699421965318 Random Forest Classification Report: precision recall f1-score support female 0.86 0.80 0.83 172 male 0.81 0.87 0.84 174 accuracy 0.83 346 macro avg 0.83 0.83 0.83 346 weighted avg 0.83 0.83 0.83 346
SVM	SVM Accuracy: 0.8352601156069365 SVM Classification Report: precision recall f1-score support female 0.90 0.75 0.82 172 male 0.79 0.92 0.85 174 accuracy 0.84 346 macro avg 0.85 0.83 0.83 346 weighted avg 0.84 0.84 0.83 346

Hence, it can be seen from both the accuracy and classification report that the random forest and SVM are the two best modules to construct hyperparameters on and further construct models on. This is because their accuracy levels are significantly higher than that of logistic regression and decision tree, about 0.84 and 0.83 compared to 0.79 and 0.80 respectively. Further, the classification report for these modelling techniques reflects that in precision, recall and f1-score – all three aspects are higher than for both random forest and SVM in the higher 0.80 to lower 0.90 range, while for both logistic regression and decision tree it is between 0.78 to 0.87 range. Regardless, the most important aspect is the precision in the classification report, whereby both genders – female and male – outperform in precision for random forest and SVM as compared to logistic regression and decision tree.

Therefore, given this intensive statistical analysis it is evident that hyperparameters will be tuned accordingly for SVM and random forest such that the most effective data from the dataset can be used to construct the overall classification model.

The hyperparameters were found for SVM and random forest, with the full code being stated in Appendix 3 and 4.

TABLE 7: *Summary of hyperparameter tuning for SVM and Random Forest.*

SVM	Hyperparameter tuning: using early stopping `Best Hyperparameters (SVM with Early Stopping): {'C': 10, 'break_ties': False, 'cache_size': 200, 'class_weight': None, 'coef0': 0.0, 'decision_function_shape': 'ovr', 'degree': 3, 'gamma': 'scalr', 'kernel': 'rbf', 'max_iter': -1, 'probability': False, 'random_state': None, 'shrinking': True, 'tol': 0.001, 'verbose': False}`
Random Forest	Hyperparameter tuning: using ensemble methods `Best Hyperparameters (Grid Search): {'learning_rate': 0.1, 'n_estimators': 50}`

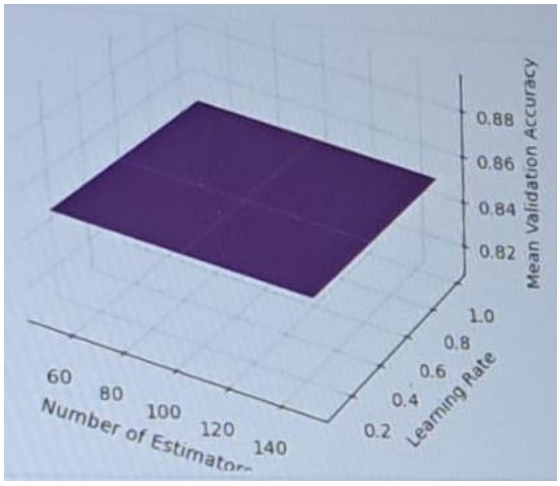
It is to be noted that SVM utilised early stopping to tune it's hyperparameters, while random forest used ensemble methods to tune it's hyperparameters. Early stopping is a form of regularisation that is "allows to find the optimal number of iterations required to build a model that generalises well to unseen data and avoids overfitting" (Scikit, 2024). Given SVM in this case specifically includes a significantly larger number of hyperparameters, early stopping is an effective method such that training happens under an effective time. The risk with many hyperparameters is that often each parameter can take a lot of time to train which delays the model being constructed. Similarly, SVM also only works in classes of two and this also adds onto the training time – more hyperparameters means more splitting required by SVM to support all the different hyperparameters (Kanade, 2024).

On the other hand, random forest utilised ensemble methods. This was much more effective for random forest as it is an ensemble learning method wherein many decision trees are constructed during training (Ashtari, 2024). As such because random forest itself is an ensemble learning method, the hyperparameter tuning using ensemble methods was highly effective and only resulted in two hyperparameters. Thereby, it will take less time for the random forest method to be trained.

Despite the differences in training time, SVM should not be negated as a model for this dataset. While random forest is clearly faster, it does not mean that it's accuracy or level of rigour is as effective as an SVM. SVM's are incredibly rigorous when it comes to training data and are industry standard machine learning models (Kanade, 2024). Random forests are typically utilised in combination with other techniques such as gradient boosting and random forest's themselves are based on decision tree (which is a machine learning model technique itself) (Ashtari, 2024). Further, SVM are also one of the most effective techniques for to model classification problems.

Hence, this can be demonstrated through visualisation, which maps all the hyperparameters and demonstrates that the despite number of hyperparameters results in a similar accuracy level is received. The full code for this is in Appendix 5.

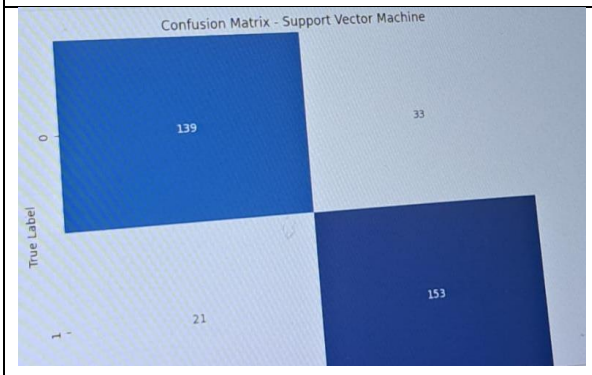
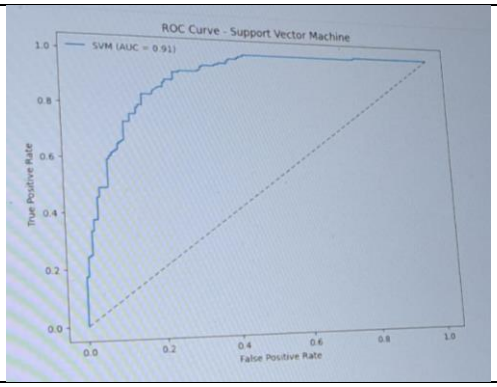
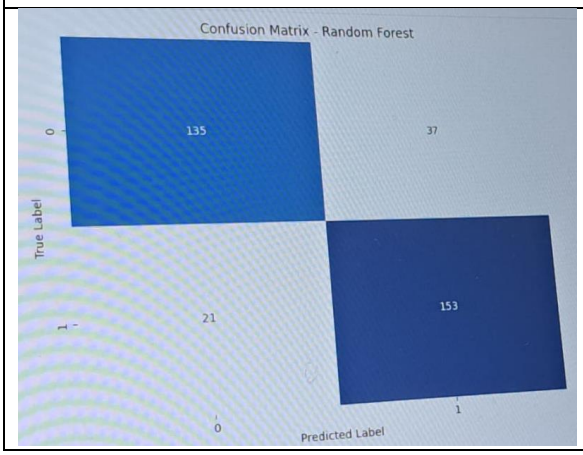
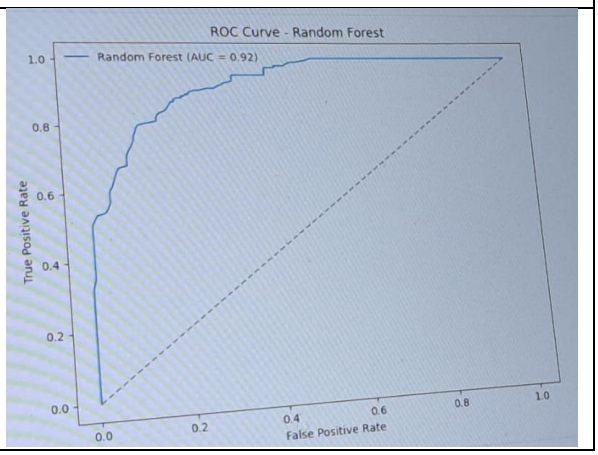
FIGURE 8: *Visualisation – hyperparameters, learning rate and mean validation accuracy*.



Hence, the number of estimates does not impact the mean validation accuracy. Therefore, despite the fact that SVM has increased amount of hyperparameters when compared to random forest it does not diminish its effectiveness as a model for this dataset.

Finally, given both the positives and negatives for both SVMs and random forests a confusion matrix and ROC curve can be constructed for both instances and the two modelling techniques can be compared in the final section of the report.

TABLE 8: *SVM versus Random Forest – Confusion Matrix and ROC Curve.*

SVM	
Confusion Matrix	**ROC Curve**
	
Random Forest	
Confusion Matrix	**ROC Curve**
	

Results

In the confusion matrix, the two sections highlighted in blue represent true positive and true negative. The top left is the true positive where the actual value is positive, and the real value is also positive. Similarly, the bottom right is the true negative where the actual value is negative, and the real value is also negative. Ideally, these two parts of the confusion matrix should be at their highest such that values that have been trained are true or false. The number of false positives and false negatives, which are on the top right and bottom left respectively should be reduced as it is not ideal to misclassify true and false values in data as otherwise.

Hence, in this case it is evident that the confusion matrix for both SVM and random forest modelling techniques are relatively similar. Similar to earlier in the report it is evident that SVM is more effective having 139 true positives, in comparison to random forest's 135 true positives. These represent the number of data points that were trained accurately and returned the same result as the raw data. It is to be recalled from Figure 7 that the training and testing data will be validated against itself to determine whether the model is effective. Similarly, SVM and random forests are equally as effective when determining true negatives demonstrating 153 values each. Therefore, it is seen that both models are more effective is disregarding values that do not fit the model as 153 is larger than both 139 and 135 for SVM and random forests respectively. True negatives are also an effective measure that random values cannot be added into the dataset and expect to be modelled. For example, if an extremely large thorax length value was to be inputted the model should not be able to give an outcome as it is out of the scope of a typical female or male fruit fly.

Thus, the total effectiveness of the model can be calculated as follows, recalling that there was a total of 346 data points that was trained:

TABLE 9: *SVM versus Random Forest – Confusion Matrix effectiveness.*

SVM	$\frac{true\ positives + true\ negatives}{total\ trained\ data\ points} = \frac{139+153}{346} = \frac{292}{346} \approx 0.8439$ $= 84.39\%$ effectiveness
Random Forest	$\frac{true\ positives + true\ negatives}{total\ trained\ data\ points} = \frac{135+153}{346} = \frac{288}{346} \approx 0.8324$ $= 83.24\%$ effectiveness

Further, the ROC curve is a graph which shows the performance of a classification model at all classification thresholds (Google, 2024). This is especially important because as noted from earlier in the report SVM and random forests have a different number of hyperparameters. This directly effects these modelling techniques' ability to classify thorax length by sex for this dataset. More hyperparameters means there is more rigorous checking no matter the number of classifiers (SVM), while less hyperparameter is a much smoother graph that approaches the false positive rate at a much slower pace (random forest). As such, it is evident that while SVM is more rigorous in its approach when classifying between the two sexes – it approaches the false positive rate much faster. False positive is a type 1 error which is the rejection of a null hypothesis when it is actually true (Kundu, 2022). Further, it is evident that random forests are not as rigorous in the early stages and overall is a smoother model which runs the risk of approaching false negative. False negative is a type 2 error which is the failure to reject a null hypothesis that is actually false (Kundu, 2022).

Therefore, though ROC is an effective measure of how closely SVM and random forest approach false positives and false negatives with increased number of classifications – it is evident that in this case the concern is with 2 classifiers (males and female). As such, early in the graph must be referred to and due to SVM being more rigorous in the early stages of classification – this is also demonstrated to be the more effective model when purely comparing the ROC outcomes.

Conclusions & Improvements

Overall, as evidenced from Table 9 and the ROC curve the SVM modelling technique is the most effective for the dataset given that the aim was to find the best classification model for thorax length by sex. It is to be noted that sex in this dataset only has two distinct values of either male or female and similarly, the whole dataset was used in its entirety without any preprocessing as there were no null values.

Regardless, there are some improvements that can be made. Most effective modelling techniques could be researched around this dataset as context is very important to ensure that the model is appropriate. This would mean the model may not take in substantially large or small thorax length values to give an outcome, as this would mean unrealistic lengths on a fruit fly. Similarly, instead of choosing common or industry-standard techniques, it could be effective to train the model by combining techniques or converting it into high-dimensional data such that more attributes can be trained, and a more effective model can be constructed. This is particularly evident in that latitude and longitude could be used as classifiers but were not in this example. Hence, rather than just distributing by sex, the regions in Eastern Australia may affect the thorax length of male and female fruit flies found in different parts.

Bibliography

Ahluwalia. (2024). Pair plots in machine learning. https://www.analyticsvidhya.com/blog/2024/02/pair-plots-in-machine-learning/#:~:text=At%20its%20core%2C%20a%20pair,patterns%2C%20trends%2C%20and%20correlations

Ashtari, H. (2024). XGBoost vs. Random Forest vs. Gradient Boosting. https://www.spiceworks.com/tech/artificial-intelligence/articles/xgboost-vs-random-forest-vs-gradient-boosting/

Chavan, R. (2020). Understanding train, test and validation dataset split in simple, quick terms. Medium. https://medium.com/@rahulchavan4894/understanding-train-test-and-validation-dataset-split-in-simple-quick-terms-5a8630fe58c8

Developers Google. (2024). ROC and AUC. https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc#:~:text=An%20ROC%20curve%20

Durgapal, A. (2020, June 30). Data preprocessing: Handling missing values in a dataset. Medium. https://medium.com/@ayushmandurgapal/data-preprocessing-handling-missing-values-in-a-dataset-5140f77d2a47

Kanade, V. (2024). Support vector machine. https://www.spiceworks.com/tech/big-data/articles/what-is-support-vector-machine/amp/

Kumara, V. (2023). Regression in machine learning: Concepts and use cases. https://builtin.com/data-science/regression-machine-learning

Kundu, R. (2022). Confusion matrix guide. https://www.v7labs.com/blog/confusion-matrix-guide

Scikit-learn. (2024). Gradient boosting with early stopping. https://scikit-learn.org/stable/auto_examples/ensemble/plot_gradient_boosting_early_stopping.html

Seaborn. (2024). seaborn.pairplot. https://seaborn.pydata.org/generated/seaborn.pairplot.html

Wikipedia. (2024). Small data. https://en.wikipedia.org/wiki/Small_data#:~:text=September%202022,small%20data%22%20is%20about%20people

Appendix

Appendix 1: Code for Pairplot of Numerical Attributes against Sex – Males versus Females

```
import matplotlib.pyplot as plt
import seaborn as sns
dataset['Thorax_length'] = pd.to_numeric(dataset['Thorax_length'], errors='coerce')

dataset = dataset.dropna(subset = ['Thorax_length'])

plt.figure(figsize = (10,6))
sns.boxplot(x = 'Sex', y = 'Thorax_length', data = dataset, palette='Set2')
plt.title('Thorax Length by Sex')
plt.xlabel('Sex')
plt.ylabel('Thorax Length')
plt.show()
```

Appendix 2: Pseudocode for Logistic Regression

1. Import necessary libraries
   - Import LogisticRegression from sklearn.linear_model
   - Import classification_report, accuracy_score from sklearn.metrics
2. Initialize the Logistic Regression model
   - logistic_model = LogisticRegression()
3. Train the model
   - logistic_model.fit(X_train, y_train)
4. Make predictions
   - y_pred_logistic = logistic_model.predict(X_test)
5. Evaluate the model
   - accuracy_logistic = accuracy_score(y_test, y_pred_logistic)
   - report_logistic = classification_report(y_test, y_pred_logistic)
6. Output the results
   - Print "Logistic Regression Accuracy:", accuracy_logistic
   - Print "Logistic Regression Classification Report:\n", report_logistic

Appendix 3: Hyperparameter tuning for SVM

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42, stratify = y)

svm_model = SVC()

X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size = 0.1, random_state = 42, stratify = y_train)

best_val_accuracy = -1
best_svm_model = None

for C in [0.1, 1, 10]:
  for gamma in ['scale', 'auto']:
    svm_model = SVC (C=C, gamma=gamma)
    svm_model.fit(X_train, y_train)

    y_val_pred = svm_model.predict(X_val)
    val_accuracy = accuracy_score(y_val, y_val_pred)

    if val_accuracy > best_val_accuracy:
      best_val_accuracy = val_accuracy
      best_svm_model = svm_model

print("Best Hyperparameters (SVM with Early Stopping)", best_svm_model.get_params())
```

Appendix 4: Hyperparameter tuning for Random Forests

```
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

rf_base_estimator = RandomForestClassifier(random_state = 42)
ada_model = AdaBoostClassifier(base_estimator = rf_base_estimator, random_state = 42)

param_grid = {
    'n_estimators': [50, 100, 150],
    'learning_rate': [0.1, 0.5, 1.0]
}

grid_search = GridSearchCV(estimator=ada_model, param_grid=param_grid, cv=5, scoring='accuracy', n_jobs=-1)
grid_search.fit(X_train, y_train)

best_params = grid_search.best_params_
print("Best Hyperparameters (Grid Search):", best_params)
```

Appendix 5: Code Visualising Hyperparameters, Learning Rate and Mean Validation Accuracy

```
[ ]  Add text cell
     import numpy as np
     import pandas as pd
     from sklearn.model_selection import GridSearchCV
     from sklearn.ensemble import AdaBoostClassifier, RandomForestClassifier
     from sklearn.metrics import accuracy_score
     import matplotlib.pyplot as plt
     from mpl_toolkits.mplot3d import Axes3D

     rf_base_estimator = RandomForestClassifier(random_state=42)
     ada_model = AdaBoostClassifier(base_estimator=rf_base_estimator, random_state=42)

     param_grid = {
         'n_estimators': [50, 100, 150],
         'learning_rate': [0.1, 0.5, 1.0]
     }

     grid_search = GridSearchCV(estimator=ada_model, param_grid=param_grid, cv=5, scoring='accuracy', n_jobs=-1)
     grid_search.fit(X_train, y_train)

     cv_results = pd.DataFrame(grid_search.cv_results_)

     fig = plt.figure()
     ax = fig.add_subplot(111, projection='3d')

     X, Y = np.meshgrid(param_grid['n_estimators'], param_grid['learning_rate'])
     Z = cv_results.pivot(index='param_learning_rate', columns='param_n_estimators', values = 'mean_test_score').values

     ax.plot_surface(X, Y, Z, cmap = 'viridis')

     ax.set_xlabel('Number of Estimators')
     ax.set_ylabel('Learning Rate')
     ax.set_zlabel('Mean Validation Accuracy')
     ax.set_title('Validation Accuracy for AdaBoost with Different Hyperparameters')

     plt.show()
```