Exploring Thorax and Wing Traits: A Classification Study of Drosophila Species

Limao Chang – 46978554

May 23, 2024

Contents

1 Introduction

Many datasets on Australian *Drosophila* (fruit fly) species have been collated and studied to investigate questions around climate adaptation, species distribution limits, and population genetics (Sandra B. Hangartner and Griffin, 2015). In this report, we investigate one such dataset (Loeschcke, Bundgaard, and Barker, 2000) which has collected various thorax and wing measurements, and focus on the classification of two such species, *Aldrichi* and *Buzzatii*, based on these traits.

In the study, the flies were taken from 5 Queensland populations: Binjour, Gogango Creek, Grandhcester, Oxford Downs, and Wahruna. They were kept in the lab for five generations at 25°C, and many thorax and wing measurements were taken on progeny of the fifth generation, which were reared at three temperature treatments (20, 25, and 30°C) (Sandra B. Hangartner and Griffin, 2015). Each measurement was replicated 3 times across 10 different vials.

2 Exploratory Data Analysis and Cleaning

2.1 Exploring the Dataset

There are $n = 1,731$ measurements and 20 columns in the original dataset. A summary of the columns is as follows:

- The majority of the columns are related to wing and thorax measurements (in mm).
- The temperature, vial, and replicate columns are related to the experimental setup of the study, and may or may not be relevant predictors for species classification. The relevance of these features will be investigated in the next subsection.
- Since the lab populations were collected in 1994, there is only one `Year_start` and `Year_end` value for all the measurements, so this feature is redundant.
- The population from which each measurement was taken is given in two forms – the location name, `Location`, and its geographical coordinates, `Latitude` and `Longitude`. These are equivalent, so we choose to ignore the `Latitude` and `Longitude` columns, as it is clearer from `Location` that it is a categorical variable.

After cleaning up the columns, there was one invalid value in `Thorax_length` and `wing_loading`. Both values belonged to the same measurement, so it was removed for the rest of the analysis.

2.2 Working with Categorical Variables

Some columns like `Sex` and `Population` are categorical variables. For our models, these need to be encoded as numeric values. The easiest way to do this is to use arbitrary integer mappings. However, we avoid doing this, as this can imply ordinal relationships that are not there. For example, if we encoded the five `Population` names as the integers 1–5, a model may compute a summary statistics like the "average location", which would not make much sense.

Instead, we use a one-hot encoding scheme to create binary indicators for each category and to avoid creating false ordinal relationships from categorical variables. For Population, this results in five new binary variables: Population_Binjour, ..., Population_Wahruna. Importantly, we also encode the replicate and vial columns in the same way. Even though they are numeric (replicates numbered 1–3 and vials numbered 1–10), these numbers are just arbitrary identifiers – really, they are just categories that happen to be numbered.

One consideration to be made with this approach is that one-hot encoding can drastically increase the number of columns, especially if each column has many unique values to encode. Here we have encoded the Sex, Population, Replicate, and Vial columns, which have 2, 5, 3, and 10 unique values respectively, so in total we have created

$$2 + 5 + 3 + 10 = 20$$

new columns as a result of the one-hot encoding (16 if we replace the original columns). This is not a significant increase in features, and we will also selectively choose features in the next subsection, so this should not be a concern. Overall, this results in 32 columns (31 of which are features), which is summarised in Table 1 below. Some of the one-hot encoded columns have been omitted for brevity.

Column Name	Data type	Description
Species	string	Species name
Population_Binjour, ..., Population_Wahruna	bool	Population indicators
Temperature	int	Rearing temperature
Vial_1, ..., Vial_10	bool	Vial indicators
Replicate_1, ..., Replicate_3	bool	Replicate indicators
Sex_male, Sex_female	bool	Sex indicators
Thorax_length	float	Thorax length (mm)
l2, l3p, l3d, lpd, l3, w1, w2, w3	float	Various wing measurements (mm)
wing_loading	float	Wing to thorax ratio

Table 1: Cleaned Dataset Columns

2.3 Feature Selection using Mean Decrease in Impurity (MDI)

Feature selection is the idea of selecting only the most relevant features for the task, while discarding redundant or inconsistent features. This can be crucial for datasets with a large number of rows and/or a large number of features (particularly if $n_{\text{features}} \gg n_{\text{rows}}$), because more features leads to more computation and a more complex model. Given two models with comparable predictive performance on the training data, we would in general prefer the simpler model, as it is less likely to overfit to unseen data (this corresponds to a lower model variance in the bias-variance tradeoff).

The dataset here is not too large, but it can still be useful to do some feature importance analysis, e.g. for interpretability of the data and results. Here we will train a random forest classifier[1] on the

[1] Later we will also use a random forest model for classification, but here we train it only for the purpose of feature selection.

training data, and rank features by their *mean decrease in impurity* (MDI).

In a decision tree, the *purity* of a node measures how “mixed” the data points within are, with respect to their classes. A node is 100% pure if all its data points belong to one class. One of the most common measures of purity is *Gini impurity*, which is a measure of how well the decision tree’s misclassification rate improves as compared to a random labelling scheme based on the distribution of classes in the data. The mean decrease in impurity for a feature then calculates how much the feature contributes to decreasing the impurity in each tree in the ensemble, weighted by the number of samples in each node, and then averaged across all the trees in the ensemble (Breiman, 2002).

In short, the idea is that **features that are more effective at distinguishing between classes will lead to larger reductions in impurity when they are used as decision nodes, and so are more important**.

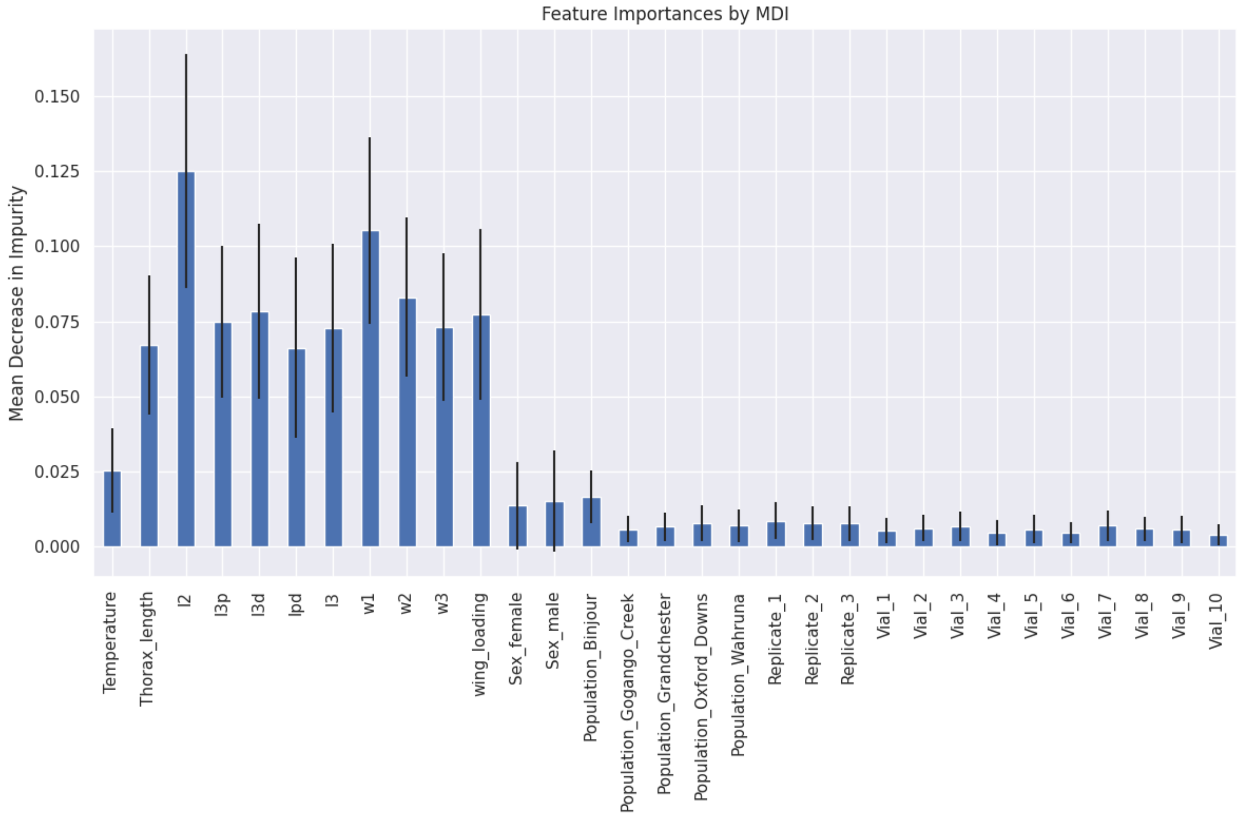

Figure 1: Feature Importances by Mean Decrease in Impurity (MDI)

Using this metric for feature importance, we can get an idea of how important each predictor is. Figure 1 shows the mean decrease in impurity for each of the 31 predictors in the dataset, with the standard deviation across the trees in the ensemble (of which there were 100) overlayed as black bars. In summary, we are left with 14 columns (13 predictors) after the feature selection process.

- The variables corresponding to the wing and thorax measurements are by far the most important predictors of species classification, which is to be expected.

- The sex and `Temperature` variables are somewhat important, so we will leave these in the dataset. However, as there are two sex categories, we can remove one (an indicator for one tells us what the indicator for the other will be).
- The replicate and vial variables have relatively small MDI values, and so are not important predictors. To save on computational efforts and model complexity, we will remove these columns.
- While most of the population variables have insignificant MDI values, `Population_Binjour` is a somewhat significant predictor. This indicates that perhaps knowing that a fruit fly comes from Binjour can be a distinguishing feature, but not so for any of the other locations. As such, we remove all of the population variables except for `Population_Binjour`.

3 Model Selection

In this section, we explore the use of several models for the species classification problem, and evaluate their results on a test dataset. 70% of the data was used for training (the same data as was used for feature selection), and 30% for testing.

3.1 Logistic Regression

A logistic regression classifier is like a linear regression model, but applies a sigmoid to the output to constrain its range to $[0, 1]$. For the two-class classification problem, the classifier will simply pick the class corresponding to the highest probability.

3.1.1 Training

The main consideration for logistic regression is whether we choose to apply regularisation when training, and if so, which regularisation scheme we choose.

Regularisation is a technique used to prevent overfitting by penalising large coefficients to keep them small. For a logistic regression model, the most common choices are the L_1 penalty term and the L_2 penalty term. The main difference is that L_1 regularisation tends to result in sparse solutions; that is, it is more likely to set particular coefficients to zero. This can be viewed as a sort of feature selection, and can so be helpful for interpretability. The L_2 penalty penalises larger coefficients more heavily than it does smaller ones, so it tends to result in solutions with smaller (but not zero) coefficients on average.

We will refer to the three models as the “no-reg” model, the L_1 model, and the L_2 model. To select the optimal value of λ for the L_1 and L_2 models, we perform stratified 5-fold cross validation (which ensures the folds have roughly the same proportion of species types). Figure 2 displays the cross validation mean accuracy scores for a range of λ values for both the L_1 and L_2 models. For the L_1 model, the optimal value was $\lambda = 0.2$, and for the L_2 model, the optimal value was $\lambda = 0.1$. Interestingly, the mean accuracy score decreases as the regularisation strength λ increases for both models. This suggests that this problem may not require much (if any) regularisation at all.

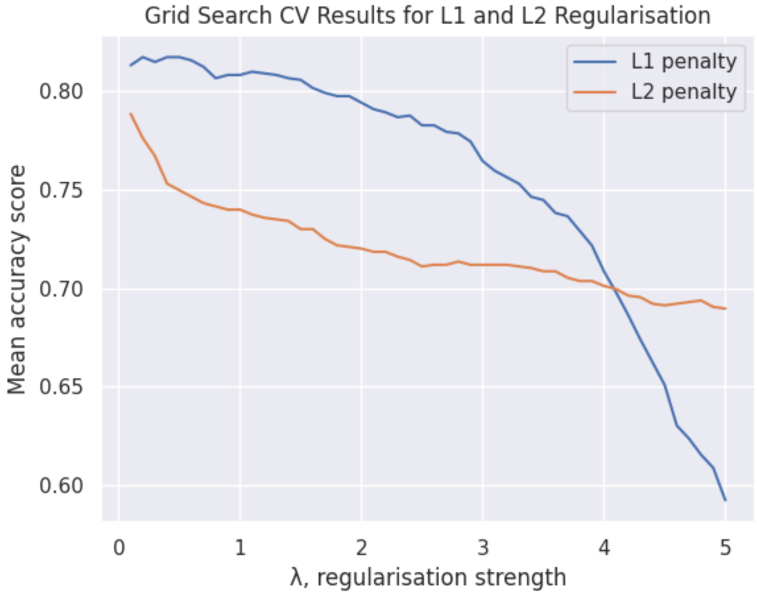

Figure 2: Grid Search Cross Validation Results for L_1 and L_2 Penalty Models

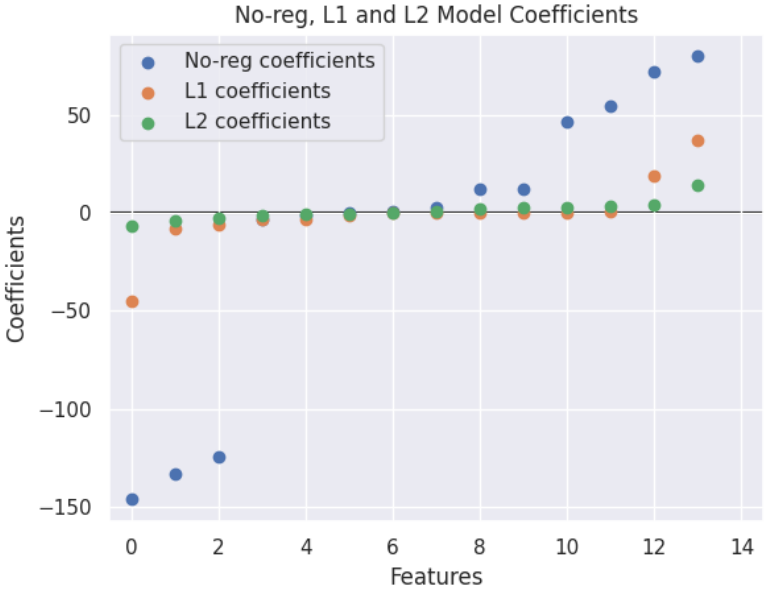

Figure 3: Sorted Coefficients of Logistic Regression Models

The (sorted) feature coefficients after training for each model are given in Figure 3. As expected, the unregularised model has larger (in magnitude) coefficients than the regularised models. The L_2 model penalises large coefficients, so it has comparatively small (in magnitude) coefficients. It is a bit difficult to tell from the graph, but the L_1 model has zeroed the coefficients corresponding to the `l3p`, `lpd`, `l3`, and `w3` features, which are all wing measurements. In comparison, neither the no-reg nor the L_2 model had any zero coefficients.

3.1.2 Results and Discussion

The results are tabulated in Table 2. The no-reg model and the L_1 model perform very similarly on the training set, with each model attaining around 82% training accuracy. The L_2 model performs slightly worse, attaining around 78% training accuracy. Interestingly, we note that the no-reg model performs the best on both the training and test datasets. That being said, the differences between the no-reg and L_1 models are quite small (<1% difference in both the training and test accuracy). The L_2 model however does perform noticeably worse than the other two.

This does corroborate with what we found in Figure 2 – which is that the mean accuracy score tended to get worse as regularisation strength increased, so indeed it is likely that this problem does not require regularisation.

Model	No-reg	L_1	L_2
Training accuracy	0.824	0.822	0.784
Test accuracy	0.794	0.784	0.751

Table 2: Performance of Logistic Regression Models

The confusion matrices (on the test dataset) for all three models are given in Figure 4. The no-reg and L_1 models specifically have a very similar number of correct and incorrect classifications for each species type.

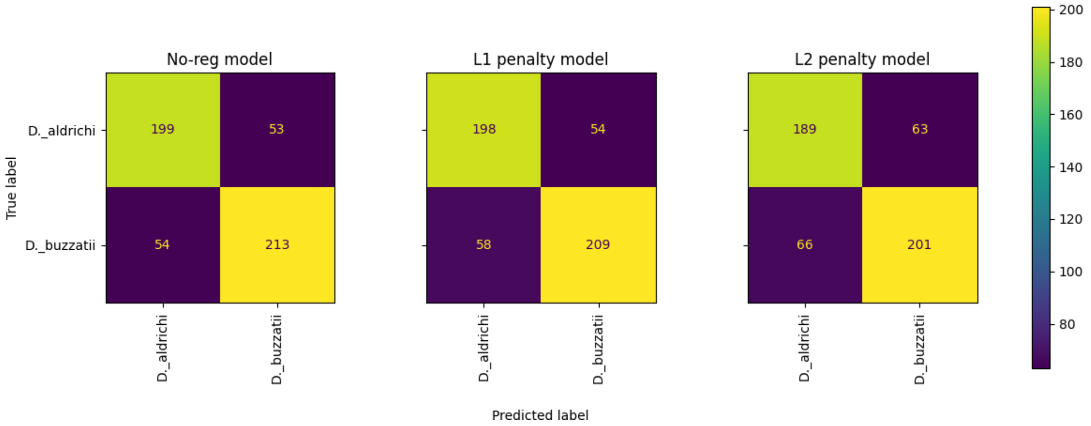

Figure 4: Confusion Matrix (test dataset) for Logistic Regression Models

3.2 Random Forest

A random forest is an ensemble model consisting of decision trees. A decision tree can be thought of as a model that predicts the classification (alternatively, the probability distribution of classifications) of a given input by answering a series of yes-no questions about the input. Each question is a decision node in the tree. A random forest's prediction is then the one with the highest mean probability across all its trees.

Usually, each tree is trained on bootstrap samples of the dataset, which creates variation in the learned trees. Random forests also employ a technique to further decrease the correlation between trees, which is to only allow each tree to consider a random subset of predictors when training and creating the decision nodes.

3.2.1 Training

During training, there are two important hyperparameters that we need to consider – the maximum depth of each tree in the forest, and the number of trees in the forest (i.e. the ensemble size).

For the former, we can use cross-validation to select the optimal depth for each of the trees.

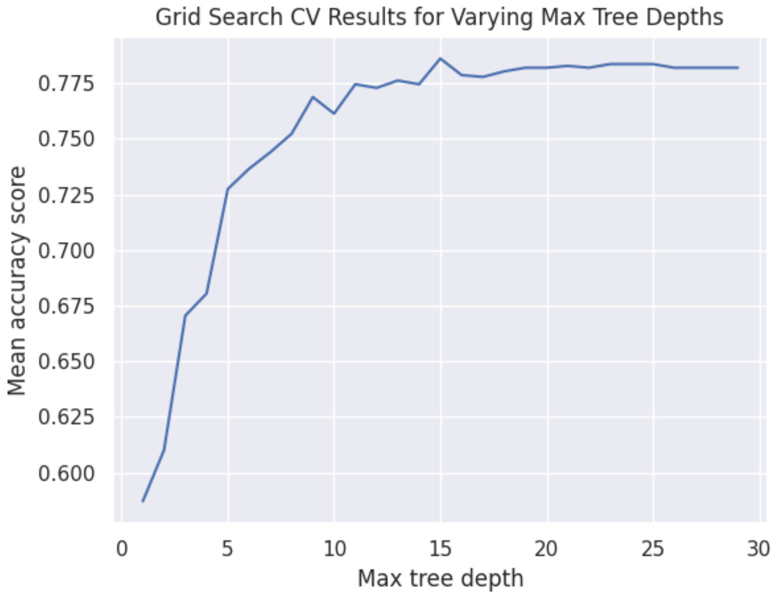

Figure 5: Grid Search Cross Validation Results for Varying Tree Depths

Figure 5 displays the cross validation mean accuracy scores for a range of max tree depths. As the max tree depth increases, the CV score also tends to increase. This is maximised at a depth of 15, which we will pick for our final classifier, as it is conveniently also a good balance between high CV score and relatively low model complexity. For the sake of comparison, we will also train a random forest classifier with a depth of 5, which will be a much simpler model.

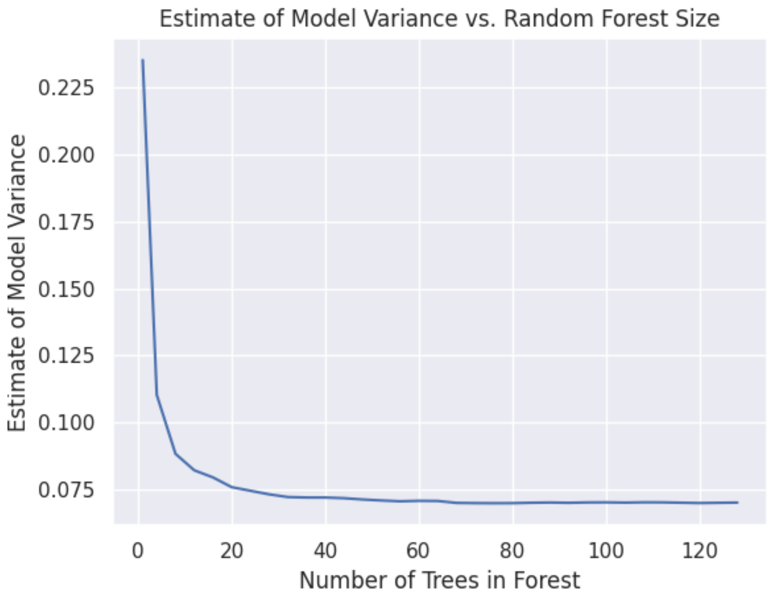

Figure 6: Estimate of Model Variance vs. Random Forest Size

For the latter hyperparameter (ensemble size), it can be shown that adding more ensemble members to the model reduces the model variance[2] without increasing the model bias (Lindholm et al., 2022). This means that having a larger number of trees in the forest does not lead to overfitting, so the main motivation for choosing a smaller number of trees is the computational cost (as each tree needs to be fit during training).

To pick an optimal number of trees, we estimate the model variance for the random forest classifier with varying numbers of trees. The results are given in Figure 6. As expected, the model variance tends to decrease as the number of ensemble members increases until it eventually plateaus. For our random forest classifiers, we will pick an ensemble of 40, as it is roughly where the variance plateaus, and so there is little benefit in picking a larger forest.

3.2.2 Results and Discussion

The results are tabulated in Table 3. Two random forest classifiers were trained on the training dataset – one with a max tree depth of 5 and one with 15. Both classifiers were trained with 40 trees in the forest. We can see from the results that both random forest classifiers exhibit similar behaviour: they performed relatively well on the training dataset but noticeably worse on the test dataset. This suggests that both models are possibly overfitting to the training data. This is almost certainly the case for the 15-depth model, as it has near perfect accuracy on the training set, but 22% less accuracy on the test dataset.

The confusion matrices for both random forest models are given in Figure 7. Both models correctly

[2] the model variance is still limited by the average correlation between the trees.

Model	max_depth=5	max_depth=15
Training accuracy	0.822	0.998
Test accuracy	0.709	0.778

Table 3: Performance of Random Forest Models

classified roughly the same number of *buzzatii* species, but the 15-depth model correctly classifies more *aldrichi* species. It appears that the 5-depth model struggles most with classifying measurements whose true species type is `aldrichi` (as it only correctly classifies ≈ 63% of these).

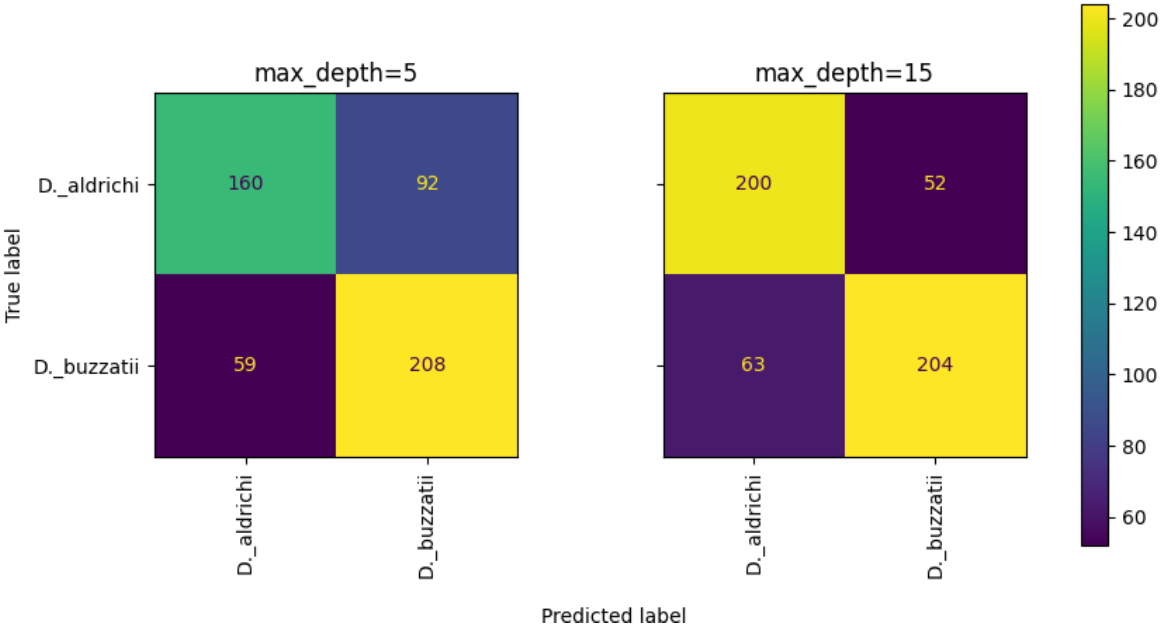

Figure 7: Confusion Matrix (test dataset) for Random Forest Models

3.3 Support Vector Machine (SVM)

A support vector machine (SVM) for the two-class classification problem aims to find a hyperplane in the feature space that best separates the data points. This is, intuitively, the hyperplane that has the largest margin between the two classes, that is, the hyperplane such that the distance to the nearest data point for each class is maximised (Lindholm et al., 2022). In the case where the data is not linearly separable, the original feature space can be transformed into a higher-dimensional feature space where it may be easier to linearly separate the data (Boser, Guyon, and Vapnik, 1992). This is essentially what we do when we make the choice of kernel in an SVM.

3.3.1 Training

For a SVM, a key consideration is the kernel type that is used, which affects the notion of similarity between data points and thus the shape of the decision boundary in the trained classifier. Another consideration is the regularisation strength λ. For this problem, we consider two choices of kernel: a Gaussian or radial basis function (RBF) kernel, and a polynomial kernel. The Gaussian kernel uses Euclidean distance as a notion of similarity, whereas the polynomial kernel uses cosine similarity on polynomial transformations of the data. Given that these kernels involve computing distances, a standardisation step was taken here to normalise each of the features before feeding it to the SVM.

Figure 8 shows the results of grid search CV for varying regularisation strengths and for the two kernels mentioned above, RBF and polynomial (with varying degrees). The effect of regularisation strength is similar to what was observed with the logistic regression model – stronger regularisation tends to be correlated with weaker CV performance. The RBF kernel and degree-1 polynomial[3] kernel performed the best, with the higher degree polynomials performing considerably worse. For both the RBF and degree-1 polynomial, the optimal regularisation value was $\lambda = 0.2$.

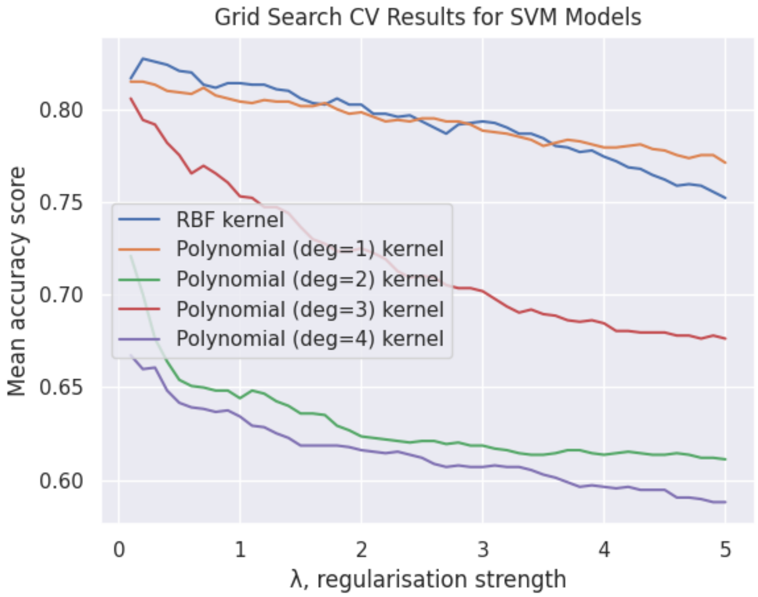

Figure 8: Grid Search CV Results for Varying SVM Kernels and Regularisation Strength λ

It is interesting that the RBF kernel performs just as well as the degree-1 polynomial kernel, since the degree-1 polynomial kernel has a linear decision boundary, whereas the RBF kernel (and the higher-degree polynomial kernels also) has a non-linear decision boundary. Although it is not possible to visualise the decision boundary without e.g. dimensionality reduction of the feature space, we know in the 2-dimensional case, the decision boundaries tend to contract around data points that are close

[3] Note that this differs slightly from a linear kernel in the `scikit-learn` implementation: the degree-1 polynomial kernel has an additional bias parameter. For this reason we will continue to refer to it as a polynomial kernel.

to each other (Varoquaux, n.d.), so one possible reason for the good performance during CV is the flexibility of the model.

3.3.2 Results

The results are tabulated in table 4. We can see that the SVM with the RBF kernel performed better across both the training and test datasets. However, the RBF kernel model seems to have overfit slightly to the training set and moreso than the degree-1 kernel model, which is as expected, given the comments in the previous subsection about the flexibility of the RBF kernel.

Model	RBF kernel	Degree-1 polynomial
Training accuracy	0.886	0.824
Test accuracy	0.821	0.782

Table 4: Performance of SVM Models

Figure 9 gives the confusion matrices for both SVM classifiers. Similarly to the previous model comparisons, both SVM classifiers are able to correctly classify roughly the same number of *buzzatii* species measurements, and the main difference between the two is the number of incorrect classifications of measurements that are *aldrichi* flies.

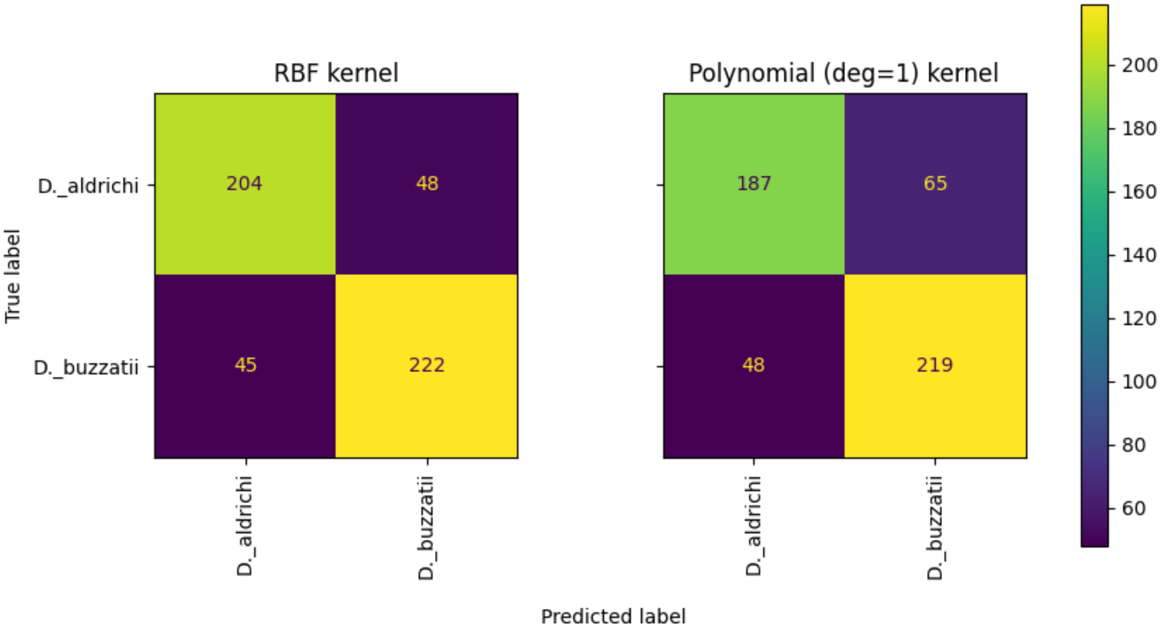

Figure 9: Confusion Matrix (test datast) for SVM Models

4 Conclusion

In this report, we explored the use of several machine learning models to classify the species of a *Drosophila* fruit fly based on various wing and thorax traits. Each of these models were able to provide some insight into the dataset, and performed reasonably well on the classification task, suggesting that the wing and thorax traits of *Drosophila* flies can be a good predictor of the species that a particular fly belongs to.

The SVM classifier with an RBF kernel performed the best on the test dataset, being the only classifier to achieve a test accuracy above 80%. Many of the other models achieved an accuracy in the high 70%s, and the logistic regression model with no regularisation in particular did quite well given the simplicity of its decision boundary (which is linear).

As noted in the training and cross-validation of several of the models, greater regularisation tended to result in worse performance. In the case of the logistic regression model, no regularisation yielded the best results on the test dataset. In the case of the SVM, a small regularisation term of $\lambda = 0.2$ yielded the best results on the test dataset. This suggests that little (if any) regularisation is required for this problem, and that the weights learned by the un-regularised models were already quite appropriate in magnitude.

For each model type explored, each variation of the model correctly classified roughly the same number of *buzzatii* species. The main distinguishing factor between the models seems to have been the performance on the *aldrichi* species – that is, the models with a higher test accuracy tended to correctly classify *aldrichi* species more often. In a sense, this means that the *buzzatii* species was "easier" to classify for each of the models in general. This may suggest that there is a stronger relationship between the various wing and thorax measurements collected for the *aldrichi* fruit flies as compared to the *buzzatii* fruit flies.

A further extension of this report could explore the use of more machine learning models. However, the results indicate that a model with a simple linear decision boundary seems to already perform quite well on the dataset, so perhaps a more complex model will not improve much on the performance already seen here. A more practical and potentially insightful extension could be to consider more of the *Drosophila* datasets (Sandra B. Hangartner and Griffin, 2015) – either individually or combined – to see if other traits of the *Drosophila* can provide more insight into the species.

References

Boser, B. E., I. M. Guyon, and V. N. Vapnik (1992). “A training algorithm for optimal margin classifiers”. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. COLT ’92. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, pp. 144–152. ISBN: 089791497X. DOI: `10.1145/130385.130401`. URL: `https://doi.org/10.1145/130385.130401`.

Breiman, L. (2002). *Manual On Setting Up, Using, And Understanding Random Forests V3.1*.

Lindholm, A. et al. (2022). *Machine Learning - A First Course for Engineers and Scientists*. Cambridge University Press. URL: `https://smlbook.org`.

Loeschcke, V., J. Bundgaard, and J. Barker (2000). “Variation in body size and life-history traits in Drosophila aldrichi and D. buzzatii from a latitudinal cline in eastern Australia”. English. In: *Heredity* 85, pp. 423–433. ISSN: 0018-067X.

Sandra B. Hangartner Ary A. Hoffmann, A. S. and P. C. Griffin (2015). “A collection of Australian Drosophila datasets on climate adaptation and species distributions”. English. In: *Sci Data*. DOI: `10.1038/sdata.2015.67`. URL: `https://doi.org/10.5061/dryad.k9c31`.

Varoquaux, G. (n.d.). *Plot classification boundaries with different SVM kernels*. `https://scikit-learn.org/stable/auto_examples/svm/plot_svm_kernels.html`.