Palmer Penguins Report

1.1 Introduction

The work carried out in this report is done on the *palmerpenguins* dataset [3]. A subset of the data, containing 6 of 8 attributes (as shown in Table 1.1), and 333 of 344 items (excluding items with 1 or more null attributes) was used for machine learning (ML) classification of the penguin species.

Table 1.1: *palmerpenguins* attributes and range of values used for ML.

Attributes	**Values**
Species	Adelie, Chinstrap, Gentoo
Bill Length (BL) / mm	32.1 - 59.6
Bill Depth (BD) / mm	13.1 - 21.5
Flipper Length (FL) / mm	172 - 231
Body Mass (BM) / g	2700 - 6300
Sex	Male, Female

1.2 Exploratory Data Analysis (EDA)

The first step looked at the distribution of species as shown in Figure 1.1. The figure has two main findings: a) the number of chinstrap penguins is significantly less than the other two species and b) sex is equally balanced for all species. Given the sex attribute is balanced, it was removed for ML since it does not add any relevant information. The second step was to look at the distribution of the four continuous attributes for each of the three species. Figure 1.2 shows a pairwise plot from which there are two main findings: a) some combinations of attributes show completely overlapping clusters and b) some kernel density plots for the attributes show a bimodal distribution. The bimodal distribution can be explained due to significant differences in the mean of the attribute values between the genders in the same species.

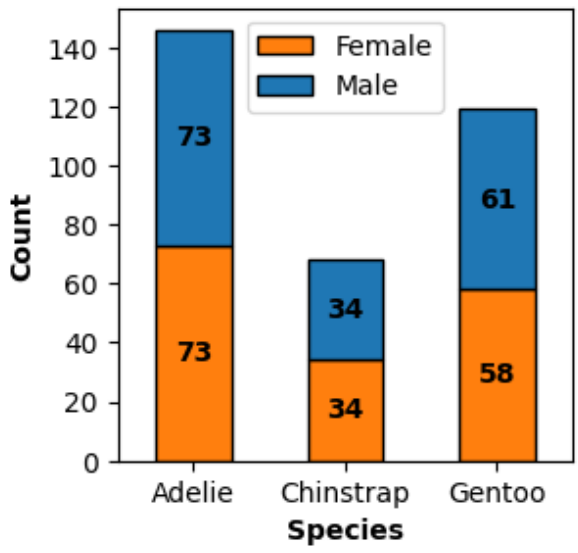

Figure 1.1: Distribution of penguin species.

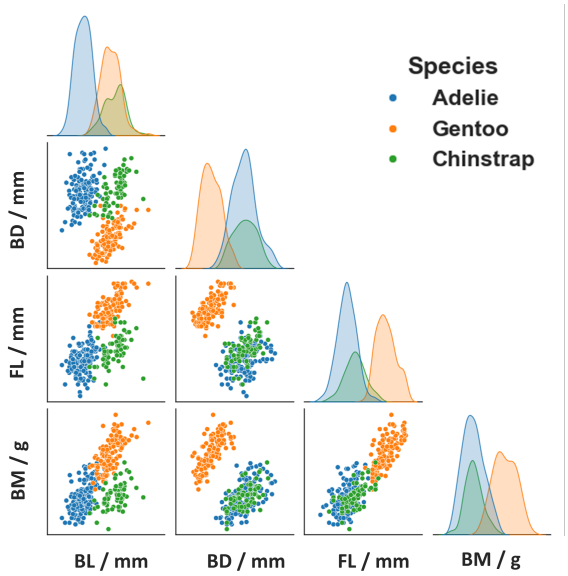

Figure 1.2: Distribution of continuous variables.

1.3 Unsupervised Learning

To conduct unsupervised learning, K Means was chosen as the clustering method. K Means was chosen because it is easy to implement, computationally efficient and has lower memory requirements [6]. To achieve that, the continuous data was transformed two-fold. First, a z-score normalization to avoid different magnitudes of the attributes affecting the distance calculations. Second, Uniform Manifold Approximation and Projection (UMAP) dimensionality reduction to ensure that all four attributes are included in two reduced dimensions. UMAP was chosen because it not only uses an unsupervised approach but also because the embedding dimension has no computational restrictions [5].

Ignoring the species labels in Figure 1.2, it can be seen that either two or three clusters are present in the scatter plots depending on the combination of attributes viewed. Therefore, the number of clusters (*n_clusters*) in K Means needs to be determined along with the UMAP hyperparameters that result in maximum separation of clusters produced in the reduced dimensions. The two main parameters used in the UMAP reduction were *densMAP* (to preserve the local density information) [7] and *n_neighbors* (number of neighbouring points used for approximations). However, only *n_neighbors* needed to optimised, given that *densMAP* is a boolean feature in the UMAP function.

The optimisation was achieved by using two 'for loops', one nested into another. The outermost loop would iterate through *n_clusters* (values include 2 and 3), and the innermost loop would iterate through the *n_neighbors*. As per the documentation [4], the suggested range of values for *n_neighbors* is between 5 to 50, with 'sensible default values' ranging from 10-15. Therefore, the range of values for exploration was 5 to 20 incrementing in steps of 5. To measure the separation of clusters, the average silhouette coefficient was recorded for each combination of *n_clusters* and *n_neighbors*.

For both *n_clusters*, the *n_neighbors* that resulted in the maximum average silhouette coefficient was 20. Figure 1.3 and Figure 1.4 show the silhouette and scatter plots for *n_clusters* 2 and 3 respectively. Given that the thickness of the silhouette plot is even amongst all the clusters and that the scatter plot visually shows three clusters, it is evident the *n_clusters* should be 3. The result in Figure 1.4 matches closely with Figure 1.5 which shows the correctly labelled scatter plots.

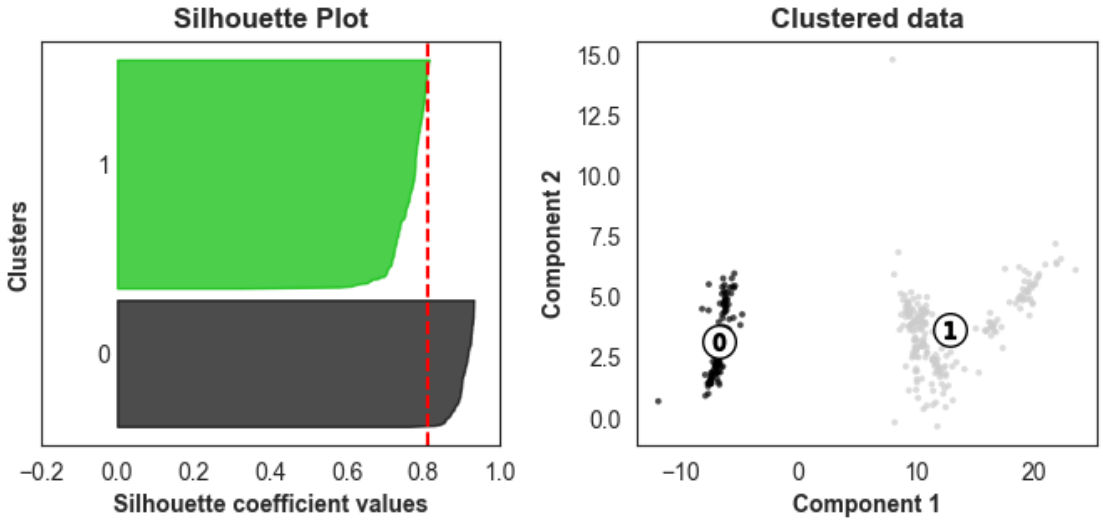

Figure 1.3: Silhouette and scatter plots for *n_clusters* = 2 and *n_neighbors* = 20.

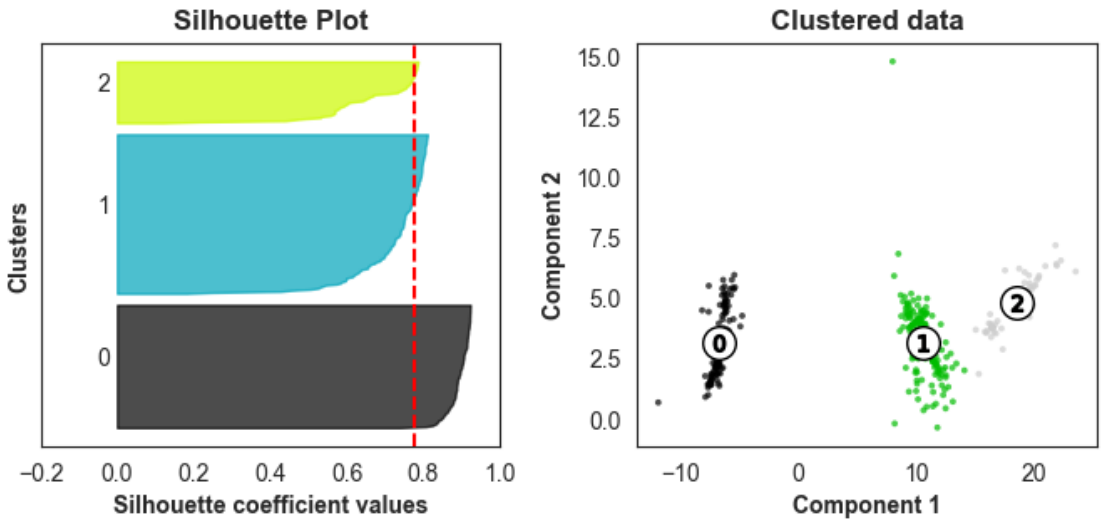

Figure 1.4: Silhouette and scatter plots for *n_clusters* = 3 and *n_neighbors* = 20.

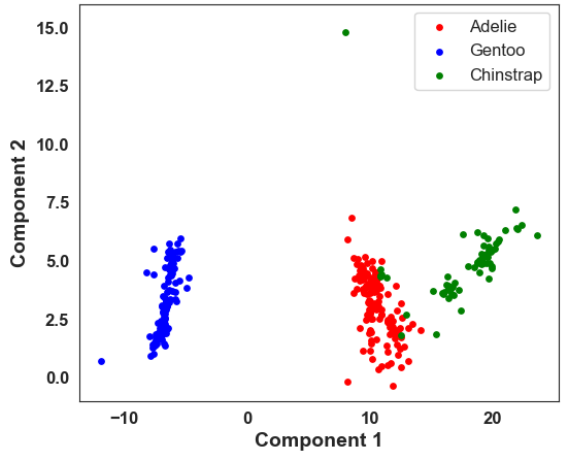

Figure 1.5: Scatter plot with correctly labelled data.

1.4 Supervised Learning

To conduct supervised learning, a suitable baseline must be established along with an appropriate performance metric. In the context of the rapid extinction of animals, misclassifying a species could lead to poor conservation management for the preservation of biodiversity. False positives would give a false sense of success whilst false negatives can influence scientists to change a successful conservation strategy. Therefore, the cost of false positives and false negatives are the same. Moreover, given that the species class in the dataset is imbalanced and all species are equally important, a macro-averaged F1 score was chosen as the performance metric. To evaluate all supervised models, the dataset was split 80/20 for training and testing respectively. Any hyperparameter tuning was done using 5 K-fold cross-validation (CV) on the training data to achieve more accurate results of the performance metric than a hold-out method [8].

For selecting a baseline, the dummy classifier was used from the sci-kit library to establish a simple rule-based baseline. Four strategies were explored: most frequent, prior, stratified and uniform. The maximum mean macro-averaged F1 score was achieved by the uniform strategy and hence was the chosen baseline.

K Nearest Neighbour (KNN) was chosen as the first supervised method since it is more complex than the baseline yet simple because it utilizes a non-parametric approach to classification [2]. KNN was set to use Euclidean distance, and so the training and testing data underwent a z-score normalization. The main hyperparameter that was tuned was the number of neighbours (k). This was done by applying 5 K-fold CV for each k ranging from 1 to 30, incrementing in steps of 1. Figure 1.6 shows the variation of the mean macro-averaged F1 score with k. The value of k was chosen by choosing the largest value that resulted in a high F1 score. Though the largest F1 score occurs at k of 2, the value of 5 was chosen since larger values of k reduces overfitting.

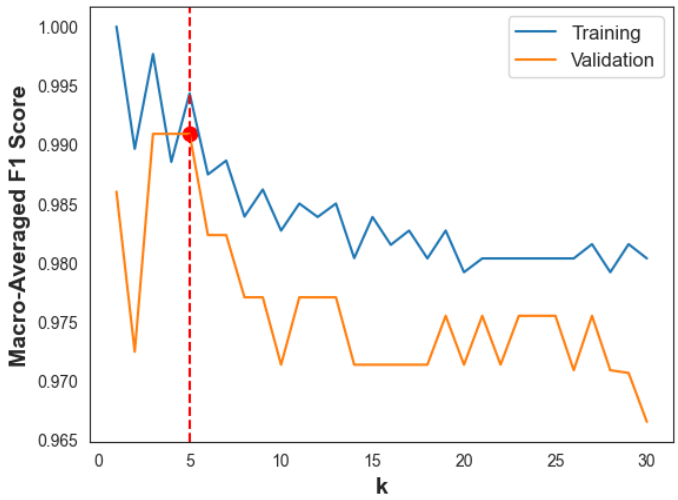

Figure 1.6: Scatter plot with labelled data.

Random Forest was chosen as the second supervised method. Random Forest was chosen over Decision Trees, since Random Forest is less prone to overfitting than decision trees [1]. Given that many hyperparameters can be tuned with a Random Forest, a GridSearch CV method was used to optimise for the highest F1 score. The key hyperparameters that were tuned were the number of trees (*n_estimators*) and the maximum depth of the trees (*max_depth*). *n_estimators* values ranged from 50 to 150 in steps of 25. *max_depth* values ranged from 2 to 5 in steps of 1 and also the default setting which would result in completely pure samples at the leaf nodes. Shannon entropy was used to measure the purity of the samples.

Table 1.2 shows the macro-averaged F1 score for all three models. Both KNN and Random Forest outperform the baseline significantly. However, comparing the KNN to Random Forest, the F1 scores for both are very similar. However, if a significantly larger dataset was used that resulted in similar F1 scores, then it would be worth comparing the speed of execution for each of the models.

Table 1.2: Supervised learning methods and results.

Model	**Parameters**	**CV F1 score**	**Test F1 score**
Baseline: Uniform	None	0.388	0.358
KNN	$k = 5$	0.991	0.945
Random Forest	*n_estimators* = 75, *max_depth* = default	0.974	0.945

References

[1] Jehad Ali, Rehanullah Khan, Nasir Ahmad, and Imran Maqsood. Random forests and decision trees. *International Journal of Computer Science Issues (IJCSI)*, 9(5):272, 2012.

[2] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. Knn model-based approach in classification. In Robert Meersman, Zahir Tari, and Douglas C. Schmidt, editors, *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, pages 986–996, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[3] Allison Marie Horst, Alison Presmanes Hill, and Kristen B Gorman. palmerpenguins: Palmer archipelago (antarctica) penguin data, 2022. R package version 0.1.0. URL: `https://allisonhorst.github.io/palmerpenguins/`, `doi:10.5281/zenodo.3960218`.

[4] Leland McInnes. Basic umap parameters — umap 0.5 documentation, 2021. [Online; accessed 07-May-2024]. URL: `https://umap-learn.readthedocs.io/en/0.5dev/parameters.html`.

[5] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[6] Laurence Morissette and Sylvain Chartier. The k-means clustering technique: General considerations and implementation in mathematica. *Tutorials in Quantitative Methods for Psychology*, 9(1):15–24, 2013.

[7] Ashwin Narayan, Bonnie Berger, and Hyunghoon Cho. Assessing single-cell transcriptomic variability through density-preserving data visualization. *Nature Biotechnology*, 2021. `arXiv:https://www.biorxiv.org/content/early/2020/05/14/2020.05.12.077776.full.pdf`, `doi:10.1038/s41587-020-00801-7`.

[8] Sanjay Yadav and Sanyam Shukla. Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. In *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, pages 78–83, 2016. `doi:10.1109/IACC.2016.25`.