Feature Engineering in MATLAB © Environment

Harti Luiz Sachser Junior
Faculty of Exact Sciences and Engineering
University of Madeira
Funchal, Portugal
2069422@student.uma.pt

Felipe Clevert
Faculty of Exact Sciences and Engineering
University of Madeira
Funchal, Portugal
2016722@student.uma.pt

***Abstract*—This paper was produced as part of the Master's in Software Engineering, ministered at the University of Madeira, Artificial Neural Networks unit. The proposed assignment was devised to go through the concept of Featured Engineering, followed by working through a few common techniques to have a rich set of suitable features for solving a machine learning problem. For this specific case study, MATLAB © software suite will be used to perform the data manipulation necessary, as well as plot the required graphs.**

Keywords—feature engineering, MATLAB, machine learning, raw data transformation, silhouette plot, clustering

I. Introduction

In the context of the curricular unit of the course above mentioned, our group was tasked with evaluating the data from the Penguin Dataset [1] using MATLAB © suite. One of the objectives of this case study is to explore this software and some of its capabilities.

The Penguin Dataset contains 344 samples of physical data, of two genders (Male and Female) of penguins, which can be categorized based on the following attributes: *species, island, culmen_length_mm, culmen_depth_mm, flipper_length_mm, body_mass_g* and *sex*. We need to make sure the dataset does not have inconsistencies and lack of data before being able to use it on a machine learning model. This is important because machine learning models can be biased towards the majority class in an imbalanced dataset, leading to poor performance in predicting the minority class. [2]

II. Development

Feature Engineering is the process of turning raw data into features to be used by machine learning. It can be quite challenging and complex, requiring deep domain knowledge to find and extract the best features. Even applying automated methods can remain an iterative and time-consuming process. In this project, we were tasked to evaluate the dataset given (Penguin Dataset) and follow a series of tasks in order to be familiar with the concept of Feature Engineering.
This dataset is based on a real case study conducted by Dr. Kristen Gorman [3] and the Palmer Station, Antarctica LTER.

A. Task 1- Importing dataset

Firstly, we were tasked with loading the Penguins dataset (*penguins_size.csv*) into MATLAB© (Eq. 1). This dataset contains information about the Culmen Length (CL), Culmen Depth (CD), Flipper Length (FL), Body Mass (BD) and Sex, regarding three species of penguins: Adelie, Chinstrap, and Gentoo. Also, the data set relates where these penguins are nested, on three different islands: Torgersen, Biscoe and Dream.

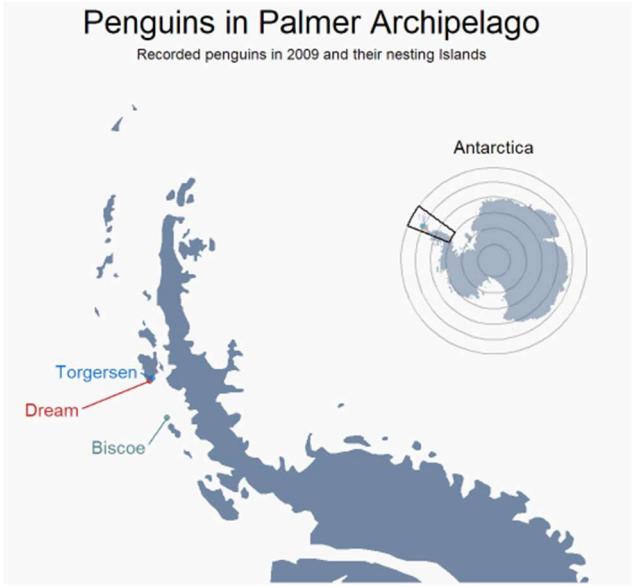

Figure 1- Palmer Archipelago

$$data\_final = readtable < filename > \quad (1)$$

Using Eq (1) on *penguins_size.csv* we generate a table and store it under *data_final:*

344x7 table

	1 species	2 island	3 culmen_length_mm	4 culmen_depth_mm	5 flipper_length_mm	6 body_mass_g	7 sex
1	'Adelie'	'Torgersen'	39.1000	18.7000	181	3750	'MALE'
2	'Adelie'	'Torgersen'	39.5000	17.4000	186	3800	'FEMALE'
3	'Adelie'	'Torgersen'	40.3000	18	195	3250	'FEMALE'
4	'Adelie'	'Torgersen'	NaN	NaN	NaN	NaN	''
5	'Adelie'	'Torgersen'	36.7000	19.3000	193	3450	'FEMALE'
6	'Adelie'	'Torgersen'	39.3000	20.6000	190	3650	'MALE'
7	'Adelie'	'Torgersen'	38.9000	17.8000	181	3625	'FEMALE'
8	'Adelie'	'Torgersen'	39.2000	19.6000	195	4675	'MALE'
9	'Adelie'	'Torgersen'	34.1000	18.1000	193	3475	''
10	'Adelie'	'Torgersen'	42	20.2000	190	4250	''
11	'Adelie'	'Torgersen'	37.8000	17.1000	186	3300	''
12	'Adelie'	'Torgersen'	37.8000	17.3000	180	3700	''

Figure 2 - data_final

B. Task 2 – Balanced Dataset

From observing the table generated in the previous task, we can see there are missing values in some of the columns. In order for us to balance it, we first need to know how many cells are missing its values. This can be achieved using the following function:

$$data\_missing = ismissing(data\_final) \quad (2)$$

C. Task 3 – Replace missing numeric and string values

Now that we know the missing values and where they are located in the table, we can replace them for appropriate values. One of the techniques that could be employed in this dataset is imputation. In data analysis, it is not uncommon to

have missing data points in a dataset due to various reasons, such as data entry errors, equipment failure, or incomplete data collection. However, most machine learning algorithms cannot handle missing data, so imputation is necessary to ensure the dataset is complete before it can be used for analysis or model training.

There are several methods for imputing missing data, however we are going to be using two, first for numerical and second for strings, respectively.

- Mean/median imputation: This involves replacing missing values with the mean or median value of the feature.
- Mode imputation: This involves replacing missing values with the mode (most common value) of the feature.

The first method can be achieved with the following function:

$$means = mean(data\_final\{:,3:6\},'omitnan') \quad (3)$$

Eq. (3) computes the mean value of all numeric rows and columns (from 3 to 6), omitting *NaN* values.

For the second method, we noticed, according to Figure 3, that the male samples (168) were only larger by 3, comparing to female samples (165). Also, the missing values for the sex column were only 3.2% of the total sample. Having said that, we used mode imputation method to replace the missing values for the most common value, namely MALE.

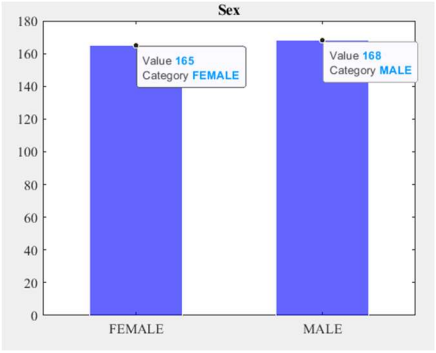

Figure 3- Female/Male occurrences

The figure below is the Penguin Dataset updated after the data analysis:

344x7 table

	1 species	2 island	3 culmen_length_mm	4 culmen_depth_mm	5 flipper_length_mm	6 body_mass_g	7 sex
1	'Adelie'	'Torgersen'	39.1000	18.7000	181	3750	'MALE'
2	'Adelie'	'Torgersen'	39.5000	17.4000	186	3800	'FEMALE'
3	'Adelie'	'Torgersen'	40.3000	18	195	3250	'FEMALE'
4	'Adelie'	'Torgersen'	43.9219	17.1512	200.9152	4.2018e+03	'MALE'
5	'Adelie'	'Torgersen'	36.7000	19.3000	193	3450	'FEMALE'
6	'Adelie'	'Torgersen'	39.3000	20.6000	190	3650	'MALE'
7	'Adelie'	'Torgersen'	38.9000	17.8000	181	3625	'FEMALE'
8	'Adelie'	'Torgersen'	39.2000	19.6000	195	4675	'MALE'
9	'Adelie'	'Torgersen'	34.1000	18.1000	193	3475	'MALE'
10	'Adelie'	'Torgersen'	42	20.2000	190	4250	'MALE'
11	'Adelie'	'Torgersen'	37.8000	17.1000	186	3300	'MALE'

Figure 4 - Data imputed

D. Task 4 – Dealing with string values

Like the issue of having missing values, no algorithm can take string values as input (Species, Island and Sex columns). One of the simplest ways of implementing a solution is to map a string to a certain numerical value and store it for future reference. MATLAB© has an in-built function called *grp2idx,* as follows:

$$[species, Vec\_species] = grp2idx(data\_final.species) \quad (4)$$

$$[islands, Vec\_islands] = grp2idx(data\_final.islands) \quad (5)$$

$$[species, Vec\_sex] = grp2idx(data\_final.sex) \quad (6)$$

Three specific vectors were created to store the relationship between the indexes and their original string values, respectively.

E. Task 5 – Clustering and Silhouette graphs

Clustering is a technique in machine learning and data analysis that involves grouping similar data points together into clusters or segments based on their intrinsic properties or similarity metrics.

The goal of clustering is to identify patterns in the data and to group similar data points into clusters so that they can be treated as a single entity or group for further analysis or processing.

For the purpose of this study, we will be using K-means clustering, one of the most commonly used algorithms, which involves dividing the data into a predefined number of clusters, with each cluster represented by its centroid. The algorithm iteratively moves the centroids to minimize the sum of squared distances between data points and their closest centroids.[4]

On the other hand, a silhouette graph is a visual representation of the quality of clustering in a dataset. It is often used in cluster analysis to evaluate the effectiveness of clustering algorithms.

In a silhouette graph, each data point is represented by a vertical line, whose height indicates the silhouette coefficient of that point. The silhouette coefficient measures how similar a point is to its own cluster compared to other clusters.

The silhouette coefficient ranges from -1 to 1, where a coefficient of 1 indicates that the point is well-matched to its own cluster, and poorly matched to neighboring clusters. A coefficient of 0 indicates that the point is equally similar to its own and neighboring clusters, and a coefficient of -1 indicates that the point is poorly matched to its own cluster, and well-matched to neighboring clusters. [5]

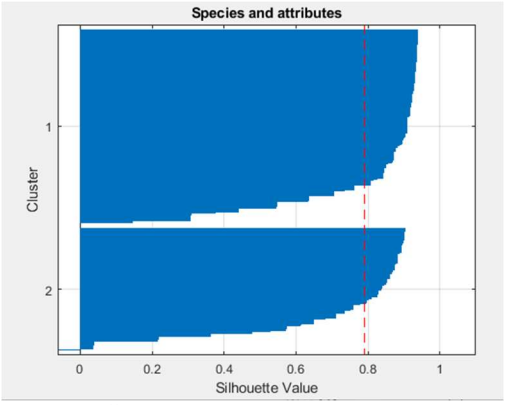

Figure 5- Silhouette Species

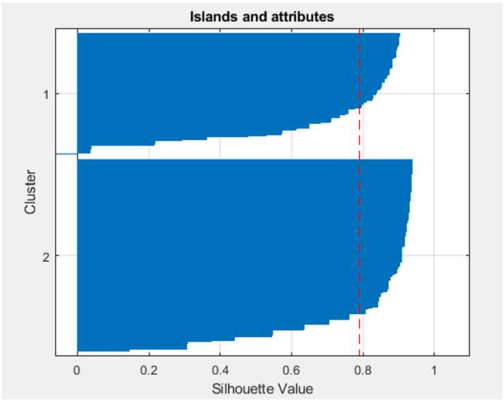

Figure 6- Islands Silhouette

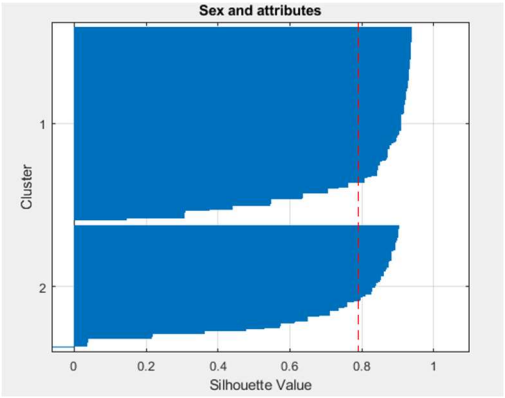

Figure 7- Sex Silhouette

F. Task 6 – Whisker Box Plot

A box plot is a type of graph that summarizes the distribution of a data set. It consists of a box that represents the middle 50% of the data, with a line inside that represents the median (or middle value) of the data set.

If any data points fall outside the whiskers, they are considered outliers and are represented as individual dots or points. Outliers are values that are significantly higher or lower than much of the data and may indicate measurement error or a rare event in the data.[6]

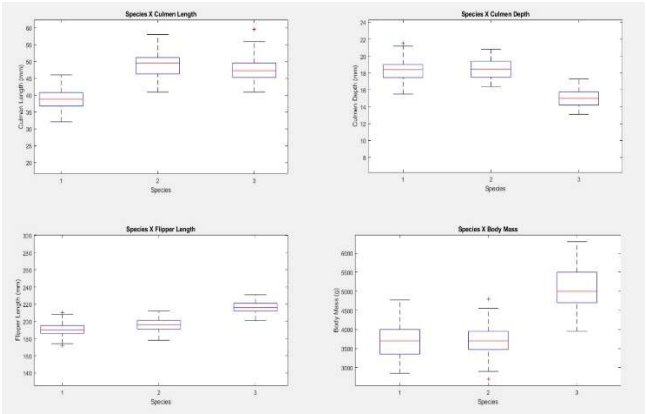

Figure 8- Species Whisker Plot

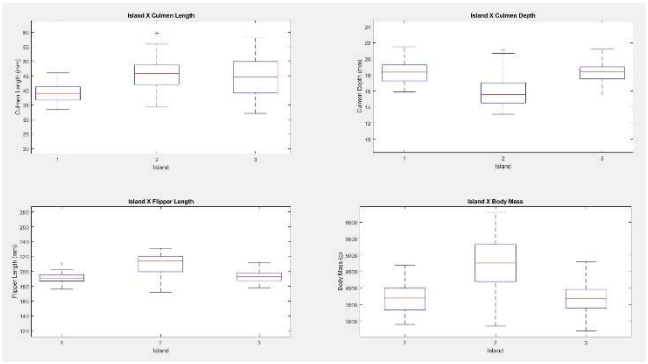

Figure 9- Island Whisker Plot

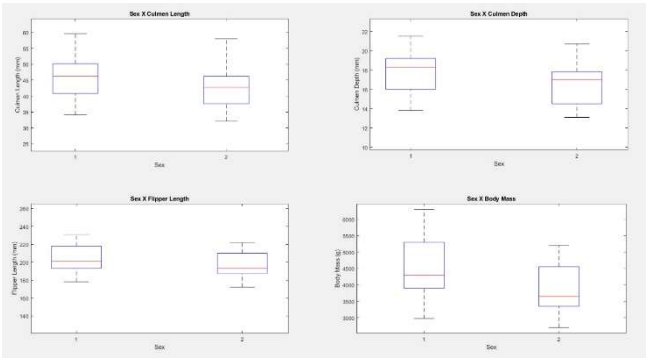

Figure 10- Sex Whisker Plot

G. Task 7 – Standard Deviation

Standard deviation is a statistical measure that represents how much the data deviates from the mean value. The standard deviation method is one of the methods that can be used for data normalization, by scaling the data based on the standard deviation.

In this process we are going to use upper and lower limits to detect outliers, on which box plots could only extract limited information about them. First step is to categorize all samples as outliers which in not covered by double standard deviation:

$$Std\_factor = 2 \tag{7}$$

$$Upper\_limit = mean(column) + std(column) * Std\_factor \tag{8}$$

$$Lower\_limit = mean(column) + std(column) * Std\_factor \tag{9}$$

$$No_{outlier} = ((column\_value > Upper\_limit)\ \&\ (column\_value > Upper\_limit)) \tag{10}$$

Running Eqs. (7-10) will give us the outliers in each column, as follows in the table below:

Table 1- Outliers

Column	Outliers
Culmen_length	6
Culmen_depth	8
Flipper_length	11
Body_mass	9

H. Task 8 – Log Transformation

Logarithmic data normalization is a type of data normalization that uses logarithmic scaling to transform data into a common format. It is particularly useful when dealing with data that has a wide range of values.

The logarithmic transformation can be done using different base values, however for this case study we are going to be using the base-10 logarithm, as follows:

$$x\_normalized = log10(x) \quad (11)$$

where x is the original data point, and $x\_normalized$ is the normalized value of x using logarithmic transformation.

We can now apply Eq. (11) to the dataset features and plot the following silhouette graphs:

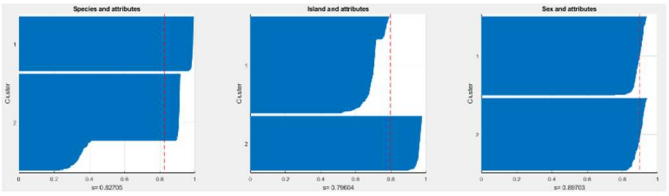

Figure 11- Log Transformation

I. Task 9 – Min-Max Scaling

Min-max normalization is one of the most common ways to normalize data. For every feature, the minimum value of that feature gets transformed into a 0, the maximum value gets transformed into a 1, and every other value gets transformed into a decimal between 0 and 1.[7]

The standard scaling for a column is defined by:

$$column = \frac{(colu \quad - mean(column))}{std(column} \quad (12)$$

The min-max scaling for a column is as follows:

$$column = min + (max - min)\frac{(column - min(column))}{max(column) - \min(column)} (13)$$

The two figures below were produced applying Eqs. (12) and (13) respectively:

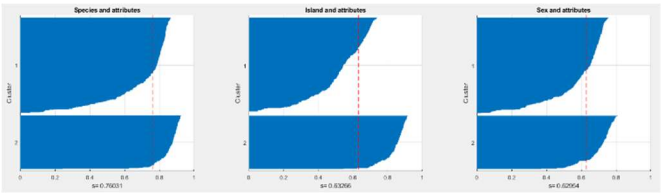

Figure 12- Standard Scaling

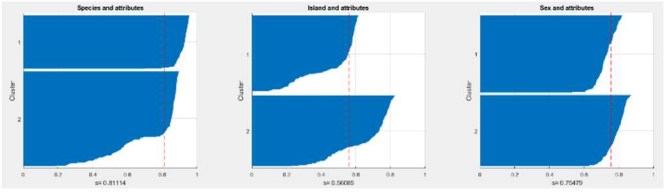

Figure 13- Min-Max Scaling

J. Task 10 – New feature

Creating new features based on current features is an important step in machine learning that can improve the accuracy and generalization of a model. This can be done by visualizing and analyzing the relationships between existing features using tools like scatter plots, correlation matrices, or principal component analysis. For example, we created features that represent the ratio of two existing features: *flipper_length/body_mass* ratio and *culmen_area (culmen_length*culmen_depth);* features that capture the interaction between two features.

For the new set of features above mentioned, we created new columns and used the logarithmic normalization, described previously, to yet again compare its silhouette plots, respectively:

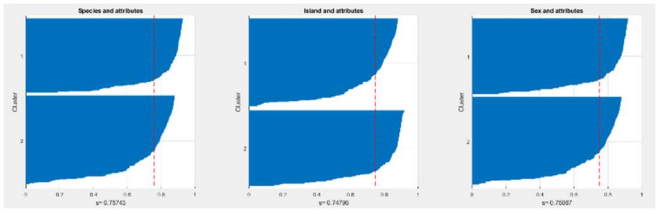

Figure 14- New feature - Body-Flipper ratio

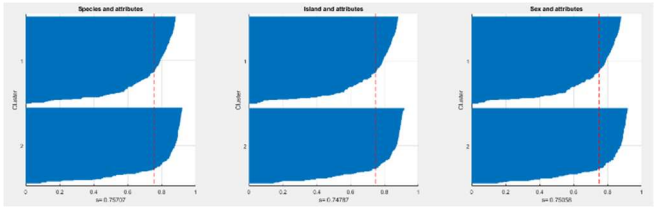

Figure 15- New feature - Culmen area

K. Task 11 – Feature Selection

In statistics, the chi-squared score (χ^2) is a measure of the difference between the expected frequencies and the observed frequencies in one or more categories of a contingency table. It is used to test the independence of two categorical variables and can also be used in feature selection to evaluate the importance of each feature in a dataset for a given classification task. Features with high chi-squared scores are considered more important for classification, as they are more strongly associated with the class variable.[8]

For this task we used a function called *'fsccchi2'* to select the best four features from a selected class variable (Species, Islands and Sex) and plot a bar graph showing their importance, in decrescent order:

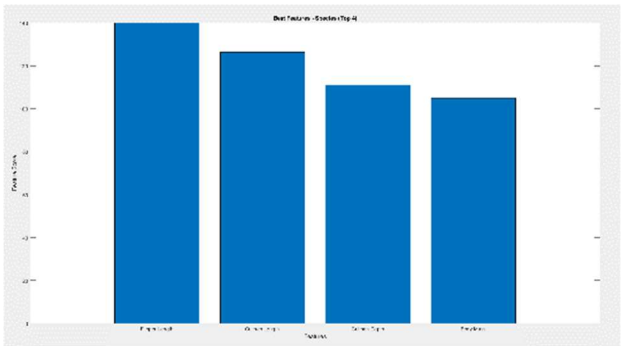

Figure 16 - Feature selection rank - Species

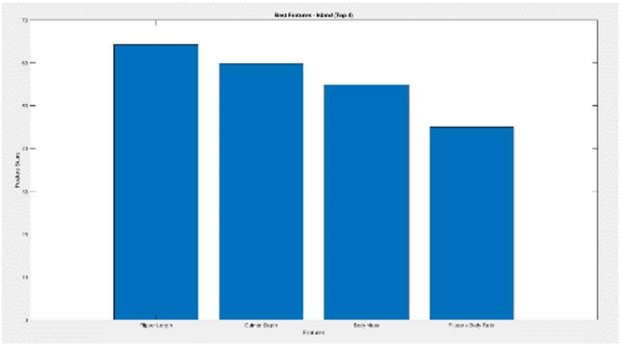

Figure 17- Feature selection rank - Islands

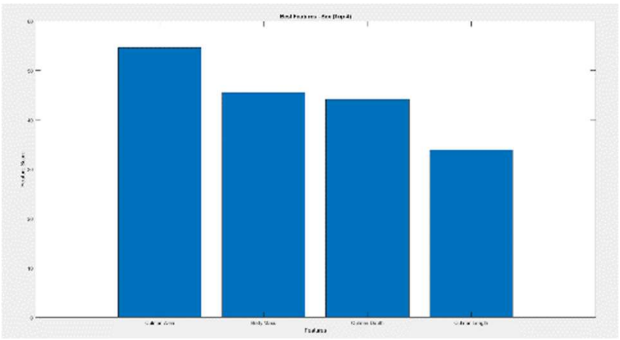

Figure 18- Feature selection rank - Sex

CONCLUSIONS

All the tasks proposed for this assignment were devised to give the students a basic understanding of feature engineering, covering a wide of concepts from statistical techniques, such as imputation, scaling, logarithm and Euclidean transformations, as well as different types of graph plots and its importance in machine learning model, regarding raw data preparation.

The choice of MATLAB© was sufficient for us to achieve all the problem-solving skills needed for this project. Overall, we concluded that the software is very powerful and resourceful for treating data, plotting all kinds of graphs and applying statistical methods. For future studies, other tools like Python could be used to perform similar tasks.

For the first part of the project (Tasks from 1 - 4), we only treated the raw data for *a posteriori* use in a machine learning model and for the software being able to handle the variables in a more efficient way. As we had missing values in our dataset, we decided to impute some known data to it based on the number of occurrences, in the case of the sex of the penguins (majority, but not by that many were MALE), and using the mean value for each column for the missing numerical values.

After that, we tested the dataset against the variable classes using the silhouette plots (Task 5) to compare how the data can be clustered together and how efficient these clusters (we only used 2 clusters for all the tasks) can represent the data classification, separating it into different and distinctive groups.

The next step (Tasks 6-7) involved detecting and deciding what to do with possible outlier values. In fact, we did detect some outliers with whisker box plots, however it was hard to tell exactly how many there were. For this, we used the standard deviation method, establishing upper and lower limits, in order to classify the data and detect the overall outliers. Detecting outliers in a dataset is important because they can significantly affect statistical analyses, machine learning models, and data visualization. Outliers can skew the mean, median, and standard deviation of the data, leading to incorrect conclusions about the underlying distribution. They can also influence regression models, causing them to fit poorly or providing false-positive results. Additionally, outliers can affect the interpretability of data visualizations and make it challenging to identify patterns and trends in the data.

When deciding what to do with outliers, it depends on the context and the cause of the outliers. If the outliers are due to measurement error or data entry mistakes, it may be appropriate to discard them. However, if the outliers represent valid data points that are truly different from the rest of the data, they may provide valuable information and should not be ignored.

Tasks (8-9) were proposed that we used methods to normalize the data and compare performance of the silhouette scores against the first set (Fig 5-7), using the same variable classes. After transforming the data using the different methods, namely Logarithmic, Euclidean (standard deviation method) and Min-Max Scaling, we concluded that the Logarithmic method was able to register a much better overall score (from s=0.796 to s=0.897) compared to the other two and to the original dataset.

The Min-Max method improved the score regarding the Species classification, however performed worse than the original dataset in the other two (Islands and Sex).

Overall, the Euclidean method performed worse than all the methods and the original dataset when comparing silhouette scores.

One of the last steps (Task 10) was to propose a new set of features, based on the current data that could correlate with the existing features, and to analyze the same way we did before, using silhouette plots and comparing its scores. We decided then to create a feature based on the ratio of the *flipper_length* to its *body_mass*, calling it *body-flipper_ratio,* and also another feature called *culmen_area,* which is the product of the *culmen_length* and the *culmen_depth.* It also needs to be noted that we used the logarithmic transformation to create these new features considering it had the best performance when clustering the data. Unfortunately, both of the features created performed slightly worse than the original dataset.

The last step consisted in categorizing the best four features, in a bar plot, from the best to worst, against the variable classes. The results are as follows:

- Species – flipper length was the best feature due to the fact that Gentoo species has much larger flipper, followed by Chinstrap and Adelie, making it easier separating them in distinctive groups;
- Islands – again the flipper length played a major role in separating the groups, since Gentoo is only found in a specific island (Biscoe), and the second best was the culmen depth, which Gentoo species varied much greater than the other two species;
- Sex – our new feature, *culmen_area* was the best feature in this case, probably because generally MALE penguins have a larger culmen than FEMALE ones for every species.

We found this study very insightful and gave us a good perspective into machine learning model preparation mixed with Data Science knowledge. Other bonus was able to enhance our skills in a powerful software like MATLAB©.

REFERENCES

[1] https://www.kaggle.com/code/parulpandey/penguin-dataset-the-new-iris

[2] https://www.analyticsvidhya.com/blog/2021/06/5-techniques-to-handle-imbalanced-data-for-a-classification-problem/

[3] Dr. Kristen Gorman, Assistant Professor, College of Fisheries and Ocean Sciences at University of Alaska Fairbanks - https://www.uaf.edu/cfos/people/faculty/detail/kristen-gorman.php

[4] https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/

[5] https://towardsdatascience.com/silhouette-coefficient-validating-clustering-techniques-e976bb81d10c

[6] https://machinelearningmastery.com/how-to-use-statistics-to-identify-outliers-in-data/

[7] https://www.codecademy.com/article/normalization

[8] Stephenson, F. H. Calculations for Molecular Biology and Biotechnology. Netherlands: Elsevier Science, 2011.