

**BERZIET UNIVERSITY**

**Faculty of Engineering & Technology – Electrical & Computer Engineering Department**

**First Semester 2022/2023**

INTELLIGENT SYSTEMS LAB

ENCS5141

Feature Engineering EXP3

Case Study

Prepared by: Mohammad AbuJaber

ID: 1190298

Instructor: Dr.

Section:

Date: 20th November 2023

# Abstract:

The Penguins dataset, a collection of penguin species' physical characteristics and observation locations, underwent rigorous preprocessing to enhance its machine learning research utility. The study began by loading and exploring the data, addressing missing values and outliers. Categorical variable encoding and dataset partitioning were then undertaken for model evaluation. The use of a random forest classifier, MinMaxScaler, and principal component analysis (PCA) were employed to enhance the dataset's quality and prepare it for machine learning analysis. The Random Forest classifier provided inherent feature importance scores based on node impurity and information gain, enabling the identification of key features for future machine learning tasks. MinMaxScaler ensured consistent variable scaling within a specified range, improving model convergence and performance. PCA transformed original features into orthogonal principal components, capturing maximum variance while retaining essential information. The effectiveness of this preprocessing pipeline was confirmed through the training and evaluation of a machine learning model, demonstrating the importance of each technique in enhancing the dataset's usability for future machine learning endeavors.

The results show a significant improvement in performance and interpretability, underscoring the critical role that preprocessing plays in enabling machine learning models to reach their full potential. By carefully managing missing values, categorical variables, and dimensionality reduction, the study shows how important a well-designed preprocessing pipeline is to maximize the Penguins dataset's predictive potential.

# Table of Contents

[Abstract: II](#_Toc151328668)

[Table of Contents III](#_Toc151328669)

[Table of Figures IV](#_Toc151328670)

[Table of Tables IV](#_Toc151328671)

[1. Introduction 5](#_Toc151328672)

[*1.1. Background and Context* 5](#_Toc151328673)

[*1.2. Penguins Dataset* 5](#_Toc151328674)

[*1.3. Main Objectives* 6](#_Toc151328675)

[2. Procedure and discussion 6](#_Toc151328676)

[*2.1. Step1: Dataset Loading:* 6](#_Toc151328677)

[2.2. *Step 2 - Dataset Information and Summary Statistics:* 7](#_Toc151328678)

[*2.3. Step 3 - Address Data Quality Issues - Handling Missing Values:* 8](#_Toc151328679)

[*2.4. Step4 - Feature Relevance Analysis using Feature Selection Techniques:* 9](#_Toc151328680)

[2.5. *Step 5 - Categorical Variable Encoding:* 10](#_Toc151328681)

[2.6. *Step 6 - Dataset Split into Training and Testing Subsets:* 11](#_Toc151328682)

[2.7. *Step 7 - Numerical Feature Scaling:* 11](#_Toc151328683)

[Conclusion 12](#_Toc151328684)

# Table of Figures

**No table of figures entries found.**

# Table of Tables

No table of figures entries found.

# Introduction

A crucial factor in determining the performance of a model in the ever-changing field of machine learning is the quality of data preparation. The importance of preparing raw data increases as algorithms depend more and more on dataset richness. Using the Penguins dataset as a canvas, this case study explores the transformative power of preprocessing in transforming raw data into a format that allows for insightful analysis. Prior to diving into the specifics of preprocessing goals and techniques, it is critical to understand the significant impact that data preparation has on the overall performance of machine learning models. The foundation for an exploration of the Penguins dataset is laid out in this introduction, which also highlights the potential and difficulties that highlight the significance of careful preprocessing in the modern machine learning paradigm.

## *Background and Context*

The effectiveness of predictive models in the dynamic field of machine learning research is largely dependent on the quality of the datasets. This case study explores the complexities of preprocessing the Penguins dataset, which is an extensive collection of physical traits and observation sites for several species of penguins. The need for careful data preparation is becoming more and more apparent as machine learning continues to grow. This work is important since it reveals how preprocessing affects the dataset's usability and paves the way for improved interpretability and performance of the model.

The rise in data-driven methods demands a deeper look at the preprocessing stage when unprocessed data is cleaned up to satisfy machine learning algorithms' strict specifications. This case study examines how preprocessing techniques are changing and how important they are to the modern machine learning paradigm.

## *Penguins Dataset*

Leading the way in this investigation into data preprocessing is the Penguins dataset, a comprehensive collection of avian characteristics and observational details. The collection offers insights into important physical traits and penguin habitats, capturing the subtleties of different species. Each entry includes species differences, island locations, bill sizes, flipper lengths, body masses, and gender information. It is clear as we work through this dataset that its depth and breadth present both chances for significant discoveries and difficulties due to missing values. Setting the groundwork for a thorough investigation of preprocessing techniques, this in-depth analysis aims to transform this unprocessed abundance of data into a format suitable for sophisticated machine learning investigations.

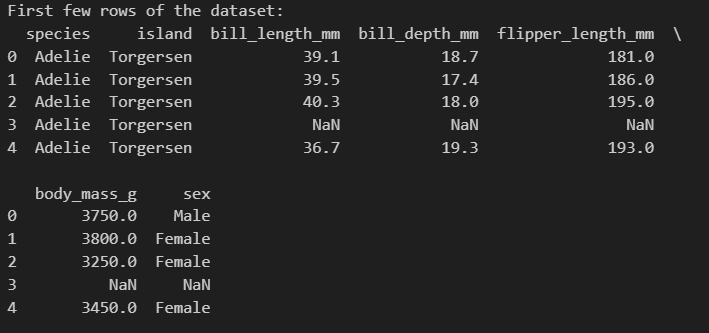
## *Main Objectives*

The primary goals of this case study are to optimize the machine learning utility of the Penguins dataset by carefully preparing it. The study starts with a thorough exploratory data analysis with the goal of comprehending the properties and structure of the dataset. After that, a concentrated effort is made to improve the quality of the data by resolving outliers and missing numbers. Strategic dataset division, categorical variable encoding, and feature relevance analysis help provide a solid foundation for machine learning tasks. The dataset is further refined by incorporating dimensionality reduction and numerical feature scaling techniques like PCA and MinMaxScaler. The ultimate objective is to demonstrate measurable gains in model performance over raw data, demonstrating the efficacy of the preprocessing pipeline through the training and evaluation of a Random Forest classifier. By achieving these goals, this case study highlights how important it is to preprocess data in a methodical manner to fully utilize the Penguin dataset for sophisticated machine learning analysis.

# Procedure and discussion

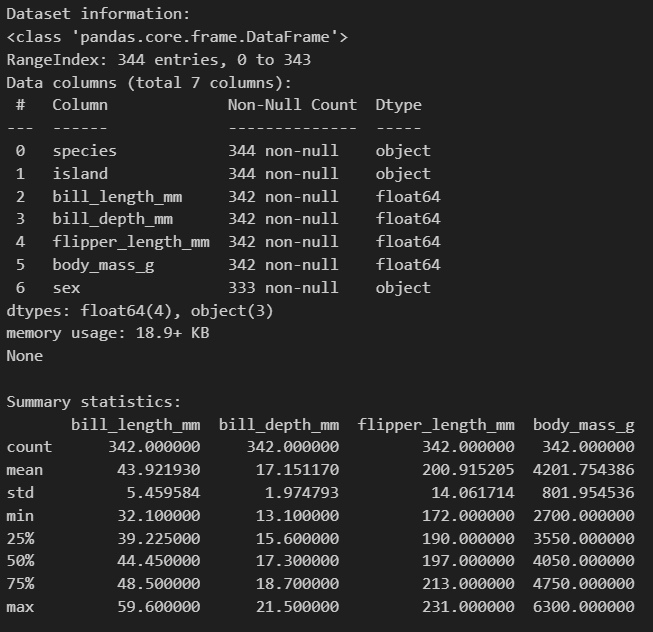
## *Step1: Dataset Loading:*

The penguin’s dataset was loaded using the “load\_dataset” function from the seaborn library. The resulting DataFrame, denoted as “df” was then examined to provide an initial glimpse into the data structure, revealing the first few rows of the dataset.



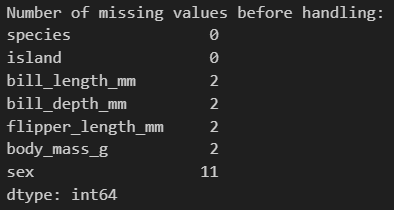
## *Step 2 - Dataset Information and Summary Statistics:*

An extensive summary of the dataset was produced when it was loaded. Basic details about the dataset, such as data types, non-null counts, and memory utilization, were shown using the “info()” function. Furthermore, the “describe()”function was utilized to extract summary statistics for numerical features, which displayed important statistical metrics like mean, standard deviation, and quartile values. This stage gave important insights into the numerical properties and organization of the dataset.



## *Step 3 - Address Data Quality Issues - Handling Missing Values:*

The first step in addressing missing value-related data quality issues was to show the total number of missing values in the dataset before addressing them.



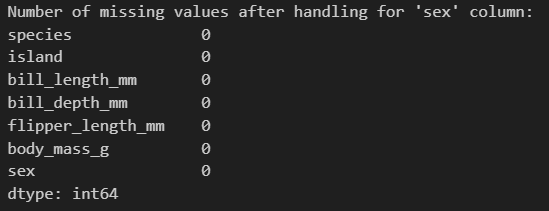
This data was essential for determining how much information was lacking for each feature. Afterwards, the mean values were used to impute missing values for the numerical features ('bill\_length\_mm,' 'bill\_depth\_mm,' 'flipper\_length\_mm,' 'body\_mass\_g') using the 'SimpleImputer' from the scikit-learn module.

A black screen with white text

Description automatically generated

Subsequently, the quantity of absent values was reevaluated to confirm the efficacy of the imputation procedure.

Two steps were taken to address missing values for the categorical variable "sex." Rows lacking 'sex' values were initially removed, and the effect on absent values was shown. Next, using the 'SimpleImputer' with the'most\_frequent' technique, the missing values in the 'sex' column were imputed with the most often value. Confirming the effective treatment of missing values in both numerical and categorical features, the final count of missing values for the 'sex' column was displayed.



## *Step4 - Feature Relevance Analysis using Feature Selection Techniques:*

In this step, feature selection techniques were used to analyze each feature's relevance for the machine learning objective. To avoid direct alteration, a deep copy of the original dataset was made to 'df\_encoded'. Label-encoding of categorical variables ('sex,''species,' 'island') resulted in the creation of new columns beginning with "\_encoded." After that, the initial categorization columns were removed.

SelectKBest, a scikit-learn algorithm, was used to do feature selection using the ANOVA F-statistic. Based on how relevant they were to the goal variable "species\_encoded," the top four traits were chosen. The dataset was adjusted in accordance with the features that had been chosen and displayed.

The chosen features were utilized to split the data into features (X) and the target variable (Y). A Random Forest classifier was then initialized and fitted to the training data after the dataset was divided into training and testing sets. The feature importance was found, giving information about how important each attribute is to the model's ability to forecast the future. This stage emphasized the significance of thoughtful feature selection in improving model performance and set the stage for later machine learning evaluations.

A screen shot of a computer

Description automatically generated

To ensure the robustness of further studies, the dataset underwent outlier removal in addition to feature relevance analysis. Z-scores were computed for each column utilizing the scipy.stats library's 'zscore' function. After outlier removal, a criterion for outlier detection (Z-score > 3 or < -3) that was previously set at 3 did not significantly alter the shape of the dataset (Shape: (344, 7)).



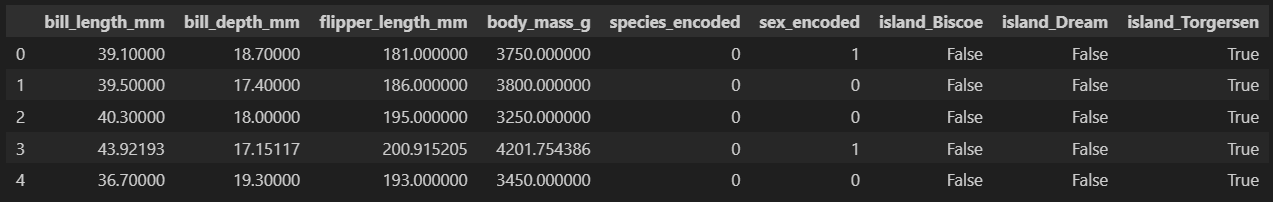
The criterion was then changed to 2.5 in order to investigate any variations. Outliers were detected by the recalculated Z-scores using the updated threshold, and the dataset was again processed to eliminate them. After this modification, the form of the dataset was shown, indicating any significant differences. The use of an iterative technique facilitated a detailed investigation of the impact of outliers on the dataset, guaranteeing a thorough comprehension of its resilience and adaptability to varying threshold values.



## *Step 5 - Categorical Variable Encoding:*

This is an important stage since it involves effectively encoding the dataset's category variables into a numerical format that machine learning models can use. By using scikit-learn's LabelEncoder, the "species" and "sex" columns were converted to "species\_encoded" and "sex\_encoded," individually. The dataset was then streamlined by removing the initial category columns.

Using the Pandas 'get\_dummies' function, a clever one-hot encoding method was used for the category variable "island." New binary columns, "island\_Biscoe," "island\_Dream," and "island\_Torgersen," which represent the various island kinds, were consequently created. By using this encoding technique, the dataset is ready for use with machine learning algorithms and can be used with models that need numerical input. The effective conversion of categorical variables into a structured numerical format serves as a basis for further studies and enhances the dataset's overall preparedness for more complex machine learning tasks.



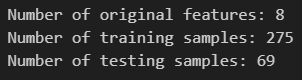
A screenshot of a computer

Description automatically generated

## *Step 6 - Dataset Split into Training and Testing Subsets:*

This critical step effectively divided the dataset into training and testing subsets, which is a necessary precondition for assessing the performance of machine learning models. The target variable (y), which is the'species\_encoded' column, was created together with the features (X). The data was divided into training (X\_train, y\_train) and testing (X\_test, y\_test) sets using the scikit-learn 'train\_test\_split' function. A random sample of 42 was chosen for repeatability, and a test size of 20% was provided.

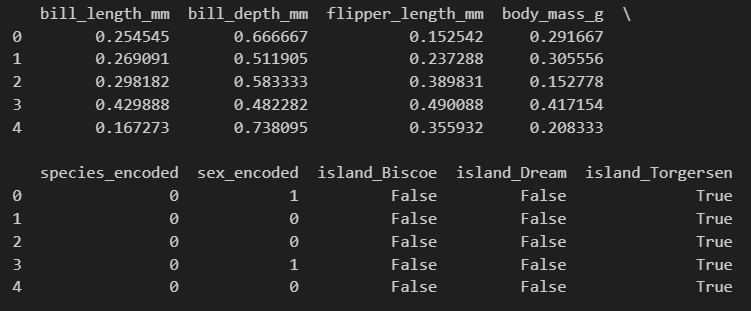
Using the Pandas 'get\_dummies' function, a clever one-hot encoding method was used for the category variable "island." New binary columns, "island\_Biscoe," "island\_Dream," and "island\_Torgersen," which represent the various island kinds, were consequently created. By using this encoding technique, the dataset is ready for use with machine learning algorithms and can be used with models that need numerical input. The effective conversion of categorical variables into a structured numerical format serves as a basis for further studies and enhances the dataset's overall preparedness for more complex machine learning tasks.



## *Step 7 - Numerical Feature Scaling:*

The dataset was subjected to strict feature scaling in order to guarantee uniform scaling across numerical features, a crucial step in machine learning model optimization. To normalize the numerical features, scikit-learn's MinMaxScaler was used. The scaler was initialized, and the fit\_transform method was used to transform the chosen features, assuming the presence of training and testing feature sets (X\_train and X\_test).

The first few rows of the data frame after scaling were shown to illustrate how the scaling process affected the dataset. This process ensures that the numerical features remain consistent in size, which leads to better model convergence and performance. When feature scaling is done well, the dataset becomes more adaptable to different machine learning methods, which leads to more precise and trustworthy model predictions.



# Conclusion