FACULTY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER ENGINEERING

INTELLIGENT SYSTEMS LAB
ENCS5141

Title: Case Study 3.1
Assignment #1

Prepared by: Ali Mohammed
ID: 1190502

Instructor: Dr. Yazan Abu Farha
Teacher Assistant: Eng. Hanan Awawdeh

Section: 1
BIRZEIT
November 2023

Abstract

The case study focuses on comprehensive data preprocessing steps applied to the Penguins dataset, aiming to prepare the dataset for machine learning analysis. The Palmer Penguins dataset comprises data from three penguin species in the Palmer Archipelago near Antarctica. Key steps in this analysis involve data visualization, descriptive statistics, handling missing data, outlier detection, feature selection, and data transformation techniques. Visualization using libraries like Matplotlib, Seaborn, and Pandas aids in understanding and communicating insights. Descriptive statistics elucidate central tendencies, variation, shape of distributions, and quantiles. Techniques for handling missing data include deletion, imputation, and creating separate categories. Addressing outliers involves identifying, choosing appropriate strategies, and reevaluating the dataset post-removal. Feature selection methods and data transformation techniques are employed to reduce dimensionality and improve model performance. The case study enhances understanding of vital data preprocessing steps, essential for robust machine learning analysis.

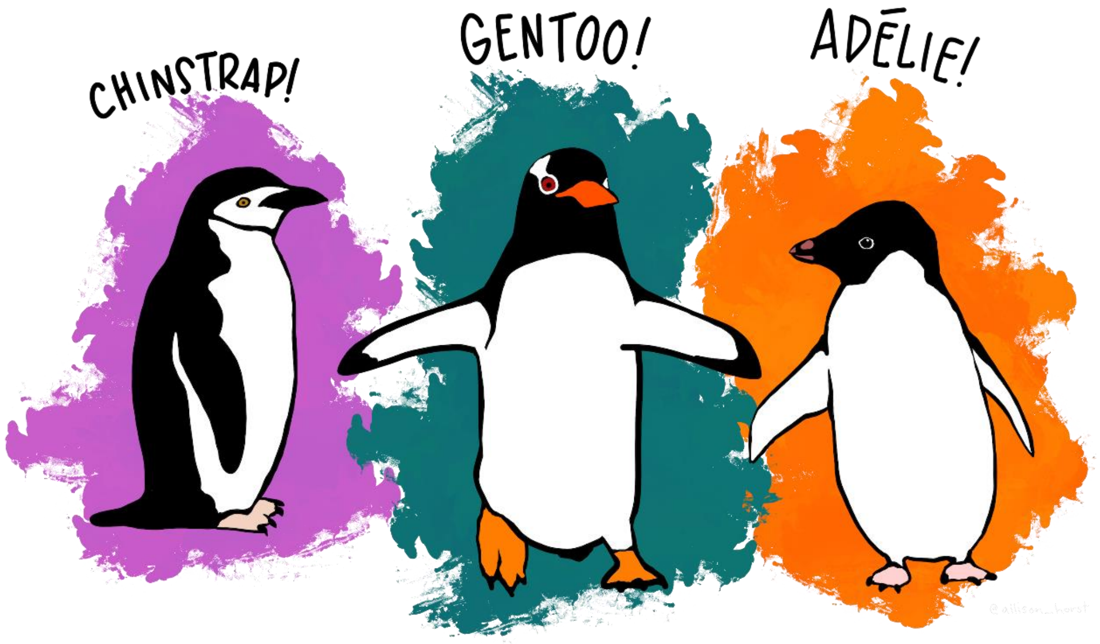
Table of Contents

List of Figures

1 Introduction

We will perform essential data preprocessing steps on the Penguins dataset. The dataset contains information about different species of penguins, including their physical characteristics and the region where they were observed. The goal is to prepare the dataset for machine learning analysis. This case study summarizes the data science part in the lab. During this case study, we will visualize the dataset, provide descriptive statistics, handle missing data and outliers, select features, transform data, and handle high-dimensional data.

1.1 Dataset



The 'Palmer Penguins' dataset has gained attention in the realm of data science as a popular dataset for analysis and machine learning tasks. This dataset contains valuable information about penguins and has been used extensively for educational and research purposes.
The Palmer Penguins dataset comprises data collected from three species of penguins inhabiting the Palmer Archipelago near the Palmer Station in Antarctica. The three species included in this dataset are Adélie, Chinstrap, and Gentoo penguins.

1.2 Data Visualization and Data Cleaning

This part shows different techniques used for data cleaning as part of an Exploratory Data Analysis (EDA). EDA plays a crucial role in comprehending and examining datasets.

1.2.1 Data visualization

Data visualization is the graphical representation of data and information through charts, graphs, maps, and other visual elements. It is a powerful way to transform raw data into a more understandable and insightful format, making it easier to identify patterns, trends, relationships, and anomalies within the data. Data visualization plays a crucial role in simplifying complex information, helping with decision-making, and effectively communicating findings.

In artificial intelligence (AI), data visualization is used to:

- **Explore and understand data**: Data visualization can be used to explore data and identify patterns and trends. This can be helpful for tasks such as data mining and machine learning.
- **Communicate insights**: Data visualization can be used to communicate insights from data to stakeholders. This can be helpful for tasks such as reporting and decision-making.
- **Generate hypotheses**: Data visualization can be used to generate hypotheses about the data. This can be helpful for tasks such as research and problem-solving.
- **Validate models**: Data visualization can be used to validate models created by AI algorithms. This can help to ensure that the models are accurate and reliable.

There are different libraries that can be used in data visualization: **Matplotlib, Seaborn, Pandas.** Also, Boxplot can be used to visualize data. Below, there are different plots using the libraries mentioned earlier and boxplot.

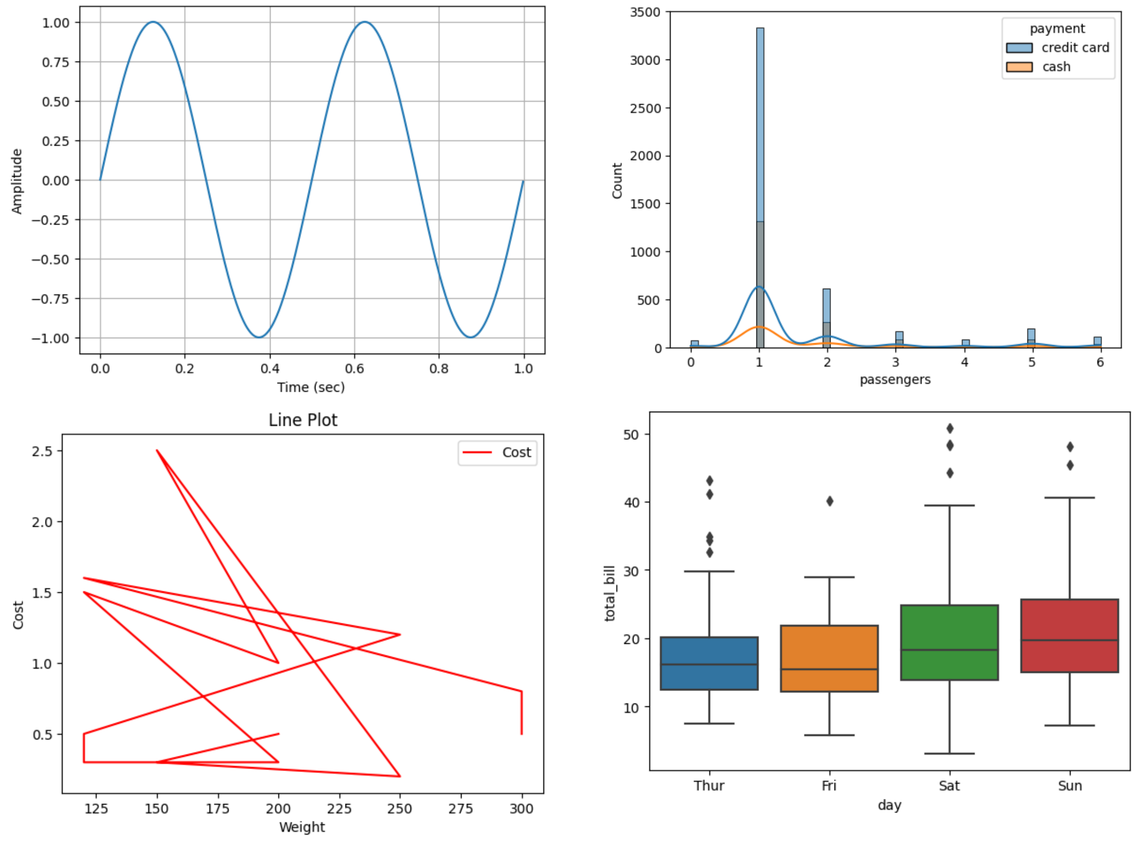


Figure 1: Plot examples using different libraries

1.2.2 Descriptive statistics

Descriptive statistics involves analyzing and summarizing data to understand its central tendencies, variability, and overall distribution. In machine learning, it plays crucial purposes like data understanding, exploration, cleaning, and preprocessing.

For data understanding and exploration, descriptive statistics offer an initial overview, resulting in understanding distribution, central tendencies, and variability. This step is

essential for identifying data patterns, anomalies, and potential issues that might affect machine learning model quality.
In data cleaning and preprocessing, descriptive statistics help in recognizing missing values, outliers, and inconsistencies. Addressing these ensures that the input data for the machine learning model is accurate and reliable.

1.2.2.1 Central tendency

The following measures are employed to assess the central tendency of a distribution of data:

- **Mean**: The average value of the data.
- **Median**: The middle value when the data is sorted.
- **Mode**: The value that appears most frequently in the data.

Pandas provides methods like **mean**(), **median**(), and **mode**() to calculate these measures.

1.2.2.2 Variation

The subsequent metrics are utilized to evaluate the dispersion of data distribution:

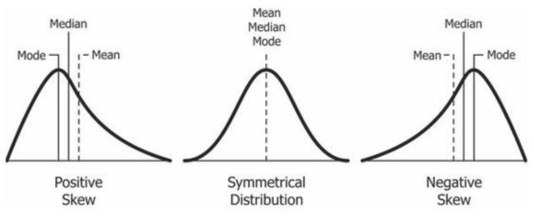
- **Variance**: A measure of how much the data points deviate from the mean.
- **Standard Deviation**: The square root of the variance, indicating the spread of data.

Pandas provides functions such as **var**() and **std**() for computing the variance and standard deviation of columns within the DataFrame.

1.2.2.3 Shape of distribution

Skewness and kurtosis are statistical measures that provide insights into the shape of a distribution:

- **Skewness**: Measures the asymmetry of the data distribution.
- **Kurtosis**: Measures the peakedness of the data distribution.



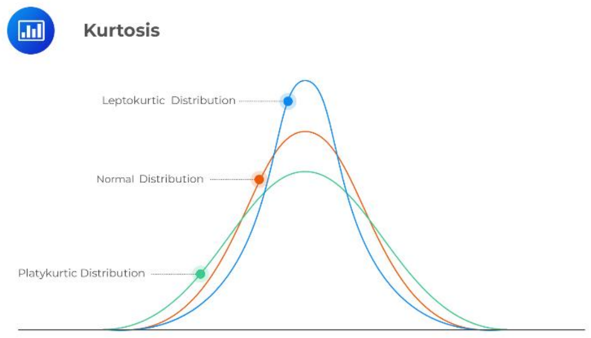


Figure 2: Demonstrating the cases of skewness and kurtosis

To calculate skewness and kurtosis for multiple columns or across the entire DataFrame, use **df.skew**() and **df.kurtosis**() without specifying a column name. These functions return Series with the skewness or kurtosis values for each column.
Note:

- If skewness is close to 0, the distribution is approximately symmetric.
- If skewness is negative, the tail is longer on the left side (left-skewed).
- If skewness is positive, the tail is longer on the right side (right-skewed).
- If kurtosis is close to 3, the distribution has similar tails as a normal distribution.

- If kurtosis is less than 3, the distribution has lighter tails and a flatter peak (a lighter-tailed distribution has a lower probability of producing values that are far from the mean compared to a distribution with heavier tails. This means that extreme values or outliers are less common in a dataset that follows a lighter-tailed distribution).
- If kurtosis is greater than 3, the distribution has heavier tails and a sharp peak (a heavy-tailed distribution is one where the tail of the distribution decays more slowly than that of a normal distribution, implying that extreme values are more likely to occur.).

1.2.2.4 Quantiles

Percentiles: Values below which a given percentage of data falls.
Interquartile Range (IQR): The range between the 25th and 75th percentiles.

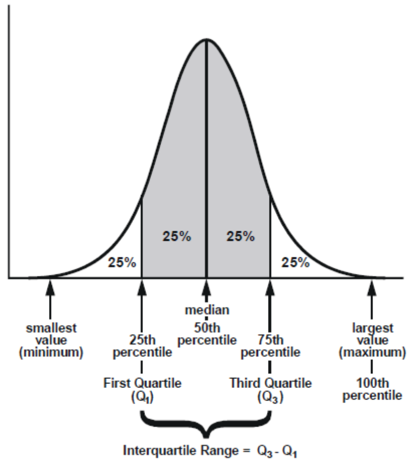
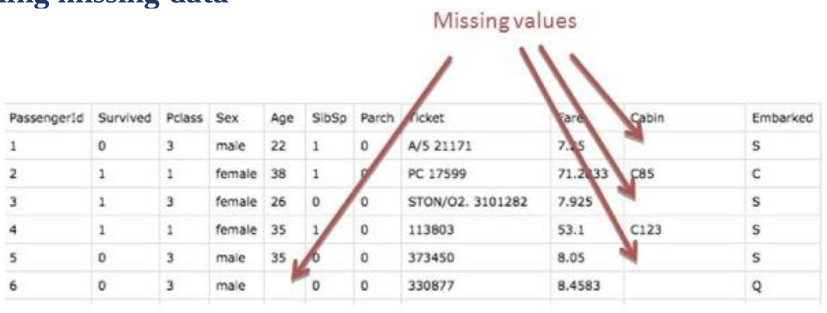


Figure 3: Demonstration of percentiles and IQR

Pandas offers the **quantile**() method for calculating percentiles. For instance, to obtain the 75th percentile, quantile(0.75) is employed, and for the 25th percentile, quantile(0.25) is used. Similarly, to compute the interquartile range (IQR), the formula **quantile(0.75) - quantile(0.25)** is applied.

1.2.3 Handling missing data


Missing values

PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	male	22	1	0	A/5 21171	7.25		S
2	1	1	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	female	26	0	0	STON/O2. 3101282	7.925		S
4	1	1	female	35	1	0	113803	53.1	C123	S
5	0	3	male	35	0	0	373450	8.05		S
6	0	3	male		0	0	330877	8.4583		Q

Figure 4: Dataset snippet with missing values

Missing data is data that is not available for one or more observations in a dataset. It can occur for a variety of reasons, such as human error, equipment malfunction, or deliberate omission.

Handling missing data is important. Missing data can have a significant impact on the quality and accuracy of a dataset. If not handled properly, it can lead to biased results and inaccurate conclusions.
There are multiple methods for handling missing data. The best method to use will depend on the specific circumstances and the goals of the analysis. Some common methods include:

- **Dropping/Deletion**: This method is used if the number of missing values in a row is relatively small, and these rows do not carry essential information for the analysis. The dropna() method in pandas is used to remove rows with missing values from the dataset.
- **Create a Separate Category**: If missing values represent a distinct category or carry specific meaning in the analysis, assign a special label or category to missing values. This makes them a distinct class in the analysis.
- **Imputation**: The imputation of missing values is used if the proportion of missing values is significant, and these data points are valuable for the analysis. There are various techniques that can be used to impute missing values:
    - **Mean, Median, or Mode Imputation**: Replace missing values with the mean, median, or mode of the observed values in the respective column.

There are different methods for handling missing values in general, and different techniques for imputing missing values. But, the mentioned methods/techniques are used in this case study and in experiments.
The impact of missing data on a dataset will depend on the following factors:

- **The amount of missing data**: The more missing data there is, the greater the impact it will have.
- **The type of missing data**: Missing data can be either missing completely at random (MCAR), missing at random (MAR), or missing not at random (MNAR). MCAR is the least harmful type of missing data, while MNAR is the most harmful.
- **The goals of the analysis**: The goals of the analysis will also affect the impact of missing data. If the analysis is sensitive to missing data, then it is important to use a method that will minimize its impact.

1.2.4 Handling outliers

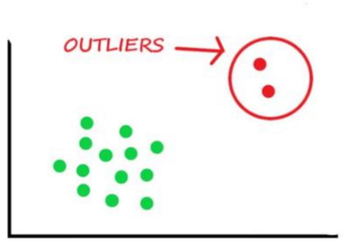


Figure 5: Data points with outliers

Removing outliers from a dataset is a critical data preprocessing step to improve the quality and reliability of the data analysis and modeling.
There is a general approach to remove outliers from a dataset:

1. **Identify the Outliers:** This step is done by visualizing the data using box plots, histograms, or scatter plots to spot potential outliers. Instead of visualization, there are statistical methods such as: z-score or the interquartile range (IQR). Data points

with z-scores beyond a certain threshold or lying outside the IQR are often considered outliers.

2. **Choose a Removal Strategy:** There are several strategies to handle outliers such as Deletion, Transformations, Capping or Winsorizing, Imputation, and Model-Based, and the choice depends on the specific use case and dataset.
3. **Apply the Chosen Strategy:** remove the outliers by implementing the chosen outlier removal strategy.
4. **Document and Justify:** Document all the changes made to the dataset, including which outliers were removed and why. This documentation is crucial for transparency and reproducibility.
5. **Reevaluate:** After removing outliers, re-evaluate the data to ensure it aligns with the analysis goals and that the removal process did not introduce any unintended biases.
6. **Sensitivity Analysis:** Perform sensitivity analysis to assess how different outlier removal methods impact your results. This can help in choosing the most appropriate method for the specific analysis.

Remember that the decision to remove outliers should be made carefully and with a deep understanding of the data. Outliers may contain valuable information or indicate data quality issues, so it is essential to strike the right balance between data integrity and data cleaning.

1.2.4.1 Detecting Outliers by Using Interquartile Range and Boxplots

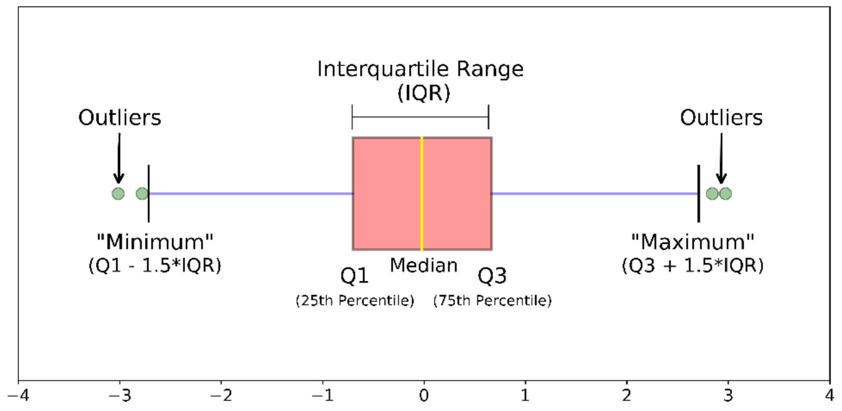


Figure 6: Different parts of a boxplot

The boxplot is a graphical representation of the distribution of a dataset that can help you detect outliers. It provides a visual summary of the data's central tendency, spread, and potential outliers. The data points beyond the whiskers are considered potential outliers.

1.2.4.2 Statistical Outlier Detection Using Z-Score

Z-Score is a statistical measure used to identify and potentially remove outliers from a dataset. By calculating the Z-Score for each data point, it can be determined how far away each point is from the mean of the dataset in terms of standard deviations. Typically, data points with Z-Scores that exceed a certain threshold are considered outliers and can be removed. Here is how to use the Z-Score method to remove outliers:

1. **Calculate the Z-Score for Each Data Point**: For each data point in the dataset, calculate the Z-Score using the formula:

$$Z_{score} = \frac{X - \mu}{\sigma}$$

Where:

- X is the data point you want to standardize.
- μ (mu) is the mean (average) of the dataset.
- σ (sigma) is the standard deviation of the dataset.

2. **Set a Threshold for Outliers:** Determine a Z-Score threshold beyond which data points are considered outliers. Commonly used thresholds are ±2, ±2.5, or ±3 standard deviations from the mean. The choice of threshold depends on the desired level of sensitivity to outliers.
3. **Identify Outliers:**
    - Data points with Z-Scores that exceed the chosen threshold are considered outliers.
    - If the Z-Score is greater than the threshold (in absolute value), it is an outlier.

1.2.4.3 Handling Detected Outliers

There are several strategies to handle detected outliers, and the choice depends on the specific use case and dataset. Here are some common approaches:

- **Deletion**: Simply, remove the outliers from the dataset. However, this approach can lead to a loss of information.
- **Transformations**: Apply mathematical transformations to the data to make it less sensitive to outliers. Common transformations include taking the logarithm, square root, or cube root of the data.
- **Capping or Winsorizing**: Cap the extreme values by setting a threshold beyond which values are considered as outliers. You can replace these extreme values with the threshold value itself or with the nearest non-outlying data point.
- **Imputation**: Instead of removing outliers, replace them with more reasonable values. This could be the mean, median, or a value predicted from a regression model. Imputation is often preferred when avoiding the loss of too much data.
- **Model-Based**: When working with predictive modeling, use robust models that are less sensitive to outliers, such as support vector machines (SVM), random forests, or robust regression techniques.

The interpretation of outliers and the choice of handling them should be done carefully, considering the context of the data and the specific goals of the analysis. Not all outliers are errors, and they can sometimes provide valuable insights into the dataset.
When the proportion of a feature outliers is small (<0.8%), the records corresponding to these outliers can be dropped.

1.3 Feature Selection and Feature Engineering

1.3.1 Feature selection

Feature selection is very important data preprocessing technique in machine learning that involves choosing a subset of the most relevant features (variables or columns) from a dataset. The goal of feature selection is to improve the performance of machine learning models by reducing the dimensionality of the data, eliminating irrelevant or redundant features, and enhancing model interpretability and generalization. There are several common feature selection methods, which can be categorized into three main types: filter methods, wrapper methods, and embedded methods.

- **Filter Methods** involves using statistical measures to identify and select the most relevant features from a dataset. Some common filter methods for feature selection include:
    - **Variance**: This method removes features that have a low variance, which means that they do not vary much across the data. This can help to remove noise and irrelevant features.
    - **Information gain**: This method measures the reduction in entropy that is achieved by knowing the value of a feature. It can be used to identify features that are most informative for predicting the target variable.
    - **Chi-squared test**: This method is used to test the independence of two variables. It can be used to identify features that are correlated with the target variable, which can be useful for classification and regression tasks.
    - **Correlation coefficient**: This method measures the linear relationship between two variables. It can be used to identify features that are highly correlated with each other, which can be removed to reduce multicollinearity.

The mentioned feature selection technique (**filter methods**) is used in the case study, while other techniques (**wrapper methods and embedded methods**) were not used.

1.3.2 Data transformation

Data Transformation is a crucial step in the data preprocessing phase of machine learning. It involves a series of operations aimed at preparing raw data into a format that is suitable for training and evaluating machine learning models. Data transformation encompasses several key processes, including scaling, normalization, discretization, and encoding, each serving its purpose in enhancing the quality and usability of the data. Here's a description of these data transformation techniques:

- **Scaling**: This method involves transforming numerical features to a common scale without changing their relative relationships. The Min-Max scaling is a commonly used method in which features are scaled to a specific range, often [0, 1]. The standardization is another type of scaling that transforms numerical features to have a zero mean and unity variance (mean=0, standard deviation=1).
- **Discretization**: This method involves converting continuous numerical data into discrete categories or bins. Some common techniques include equal-width binning (dividing the data into equal-width intervals) and equal-frequency binning (ensuring each bin contains approximately the same number of data points).
- **Encoding**: Encoding is the process of converting categorical data (text or labels) into a numerical format that machine learning models can understand. Some common encoding methods include one-hot encoding (creating binary columns for each category), label encoding (assigning a unique integer to each category), and ordinal encoding (mapping ordinal categories to numerical values).
-

1.3.3 Handling high-dimensional data

There are two prominent methods for dimensionality reduction: Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA).

Principal Component Analysis (PCA): PCA is one of the most widely used techniques for reducing the dimensionality of data. It transforms the original features into a new set of orthogonal features called principal components. These components capture the maximum variance in the data, allowing for dimensionality reduction while retaining as much information as possible.

Linear Discriminant Analysis (LDA): LDA is used when the goal is not just dimensionality reduction but also class separation. It finds a linear combination of features that maximizes the distance between different classes while minimizing the variance within each class.

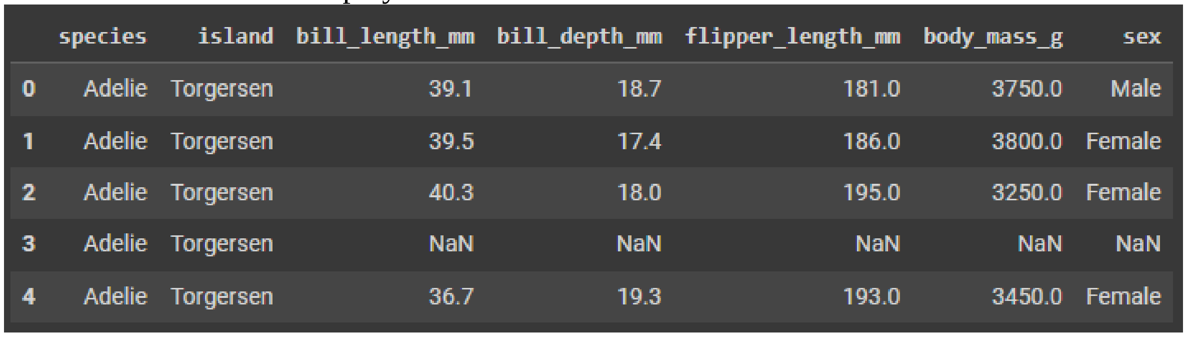
Both PCA and LDA are dimensionality reduction techniques, PCA is unsupervised and aims to capture maximum variance, whereas LDA is supervised and focuses on maximizing class separation. The choice between PCA and LDA depends on the specific goals of your machine learning task and whether you are dealing with a classification problem.

2 Procedure and discussion

This section will contain the procedure of the case study and discuss the result/s of each one.

2.1 Load the penguin's dataset

In this step, the penguin's dataset is loaded using the code snippet in the ipynb file. And, the header of the dataset is displayed.



	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	Male
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	Female
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	Female
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	Female

Figure 7: Dataset header

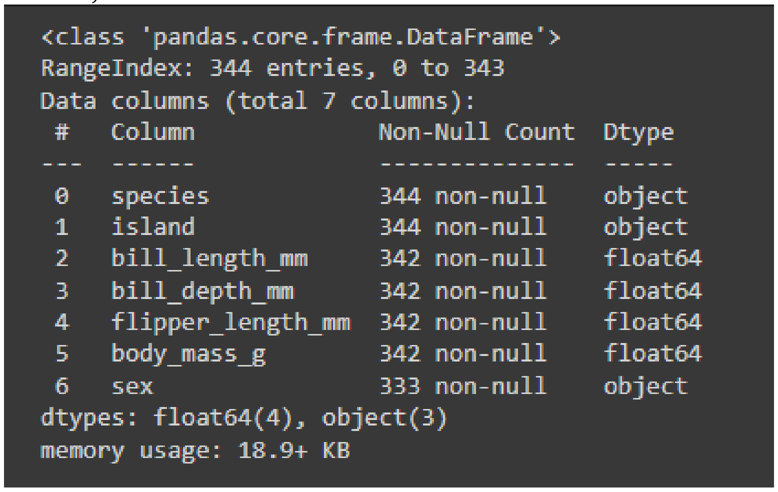
Initially, the dataset has 7 columns (features): species, island, bill_length_mm, bill_depth_mm, flipper_length_mm, body_mass_g, sex. Also, there is missing values in the dataset such as row 3.

2.2 Data exploration

The aim of this step is to understand the dataset's structure, features, summarize its statistics, and gain insights into the data.

2.2.1 Dataset structure

To understand the dataset structure, **info()** method is used. This method displays: the number of entries and columns, each column with the number of non-null entries and its type.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 344 entries, 0 to 343
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   species            344 non-null    object
 1   island             344 non-null    object
 2   bill_length_mm     342 non-null    float64
 3   bill_depth_mm      342 non-null    float64
 4   flipper_length_mm  342 non-null    float64
 5   body_mass_g        342 non-null    float64
 6   sex                333 non-null    object
dtypes: float64(4), object(3)
memory usage: 18.9+ KB
```

Figure 8: The output of info() method

The features of the dataset aret: species, island, bill_length_mm, bill_depth_mm, flipper_length_mm, body_mass_g, sex.

The number of features is 7.
The number of entries (samples) is 344.

2.2.2 Data visualization

This part is to understand the features and their relationship by visualizing them.

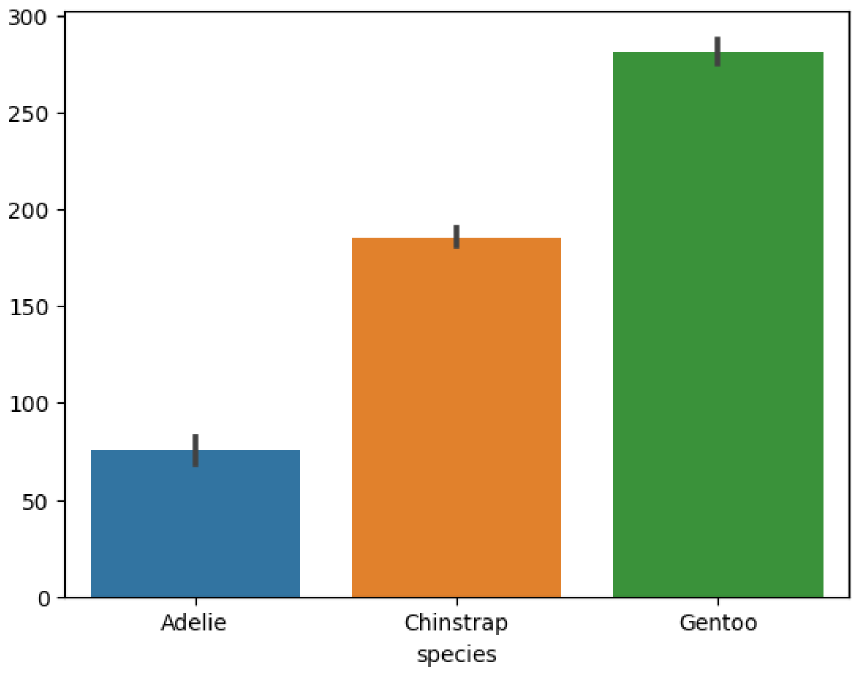


Figure 9: The distribution of species

It consists of three types: Adelie, Chinstrap, Gentoo. The most frequent one is 'Gentoo'.

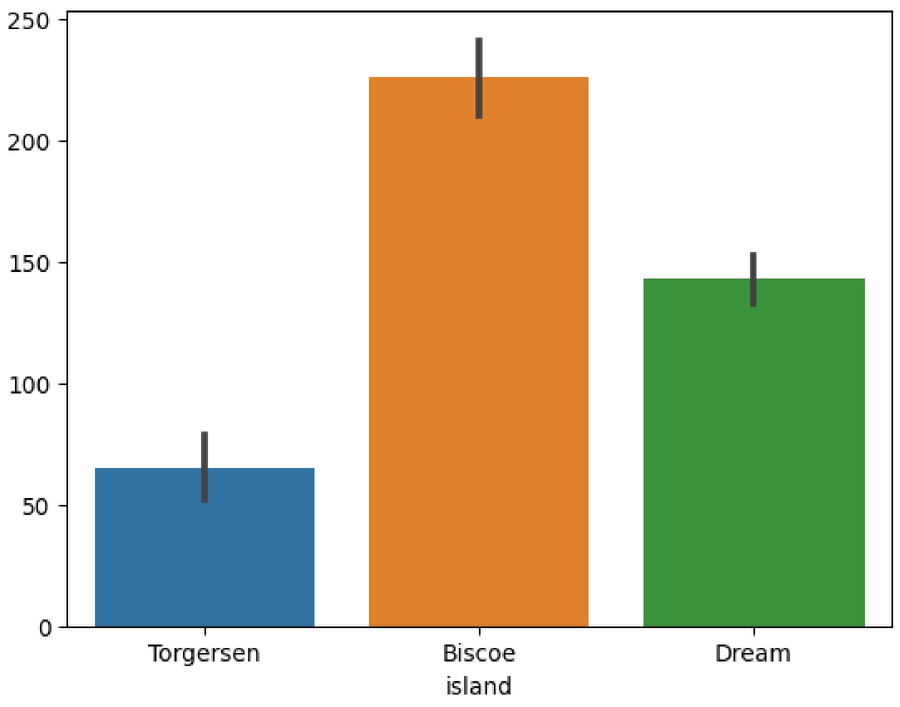


Figure 10: The distribution of island

It consists of three types: Torgersen, Biscoe, Dream. The most frequent one is 'Biscoe'.

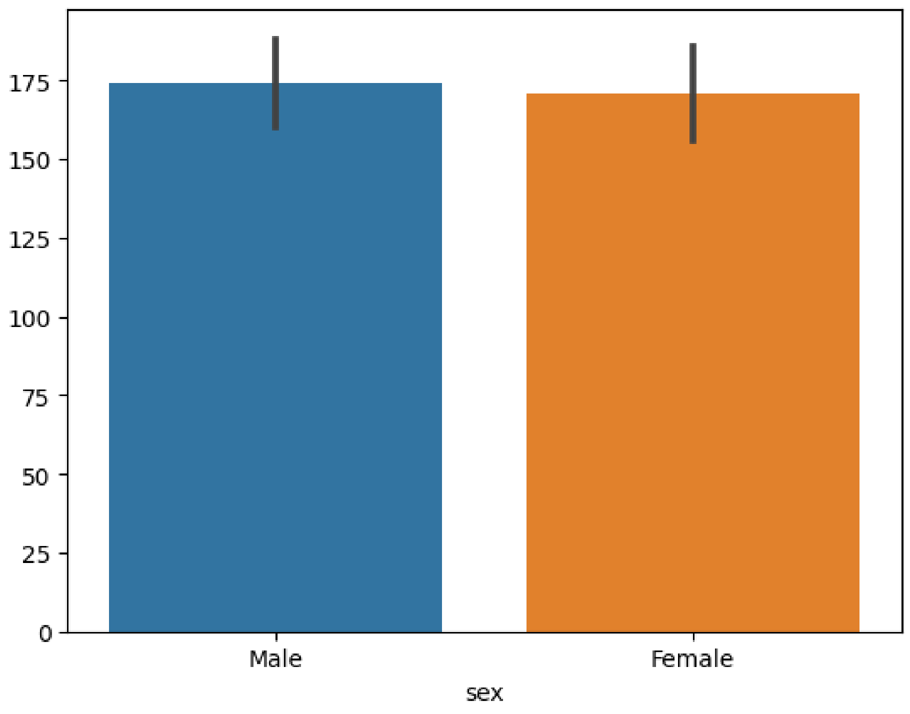


Figure 11: The distribution of sex

It consists of two types: Male and Female. The number of males is a bit greater than females.

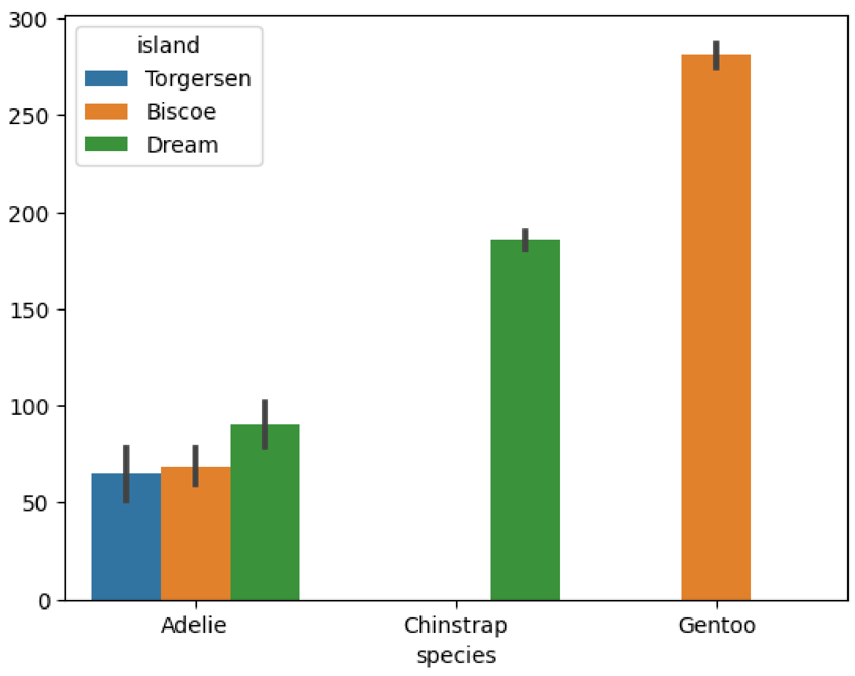


Figure 12: The distribution of species with respect to the island

The graph shows the distribution of species according to the island. It is obvious that Chinstrap lives in island Dream only. While Gentoo lives in Biscoe. As well as, Adelie live in Torgersen, Biscoe, & Dream, but mostly in Dream.

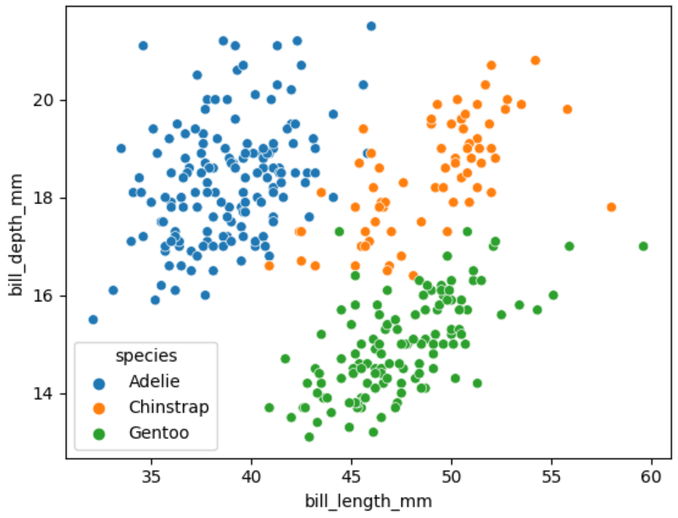


Figure 13: The relationship between bill_depth_mm and bill_length_mm

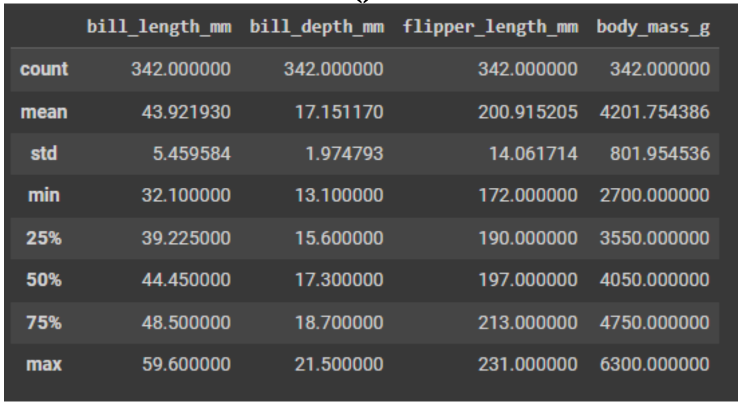
The graph shows the relation between bill_length and bill_depth with respect to the species types, it is clear that we three groups/clusters.

- **Adelie**: the bill legnth ranges 30 up to 40, and the depth from 16 - 22. Both range are (approximation).
- **Chinstrap**: the bill legnth ranges 40 up to 60, and the depth from 16 - 22. Both range are (approximation).
- **Gentoo**: the bill legnth ranges 40 up to 60, and the depth from 12 - 16. Both range are (approximation).

To shorten the report, the most important plots were introduced above. There are many plots available on the jupyter file with their disunion.

2.2.3 Descriptive statistics

In general, there is a method called **describe()** that describes the statistics of the dataset.



	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
count	342.000000	342.000000	342.000000	342.000000
mean	43.921930	17.151170	200.915205	4201.754386
std	5.459584	1.974793	14.061714	801.954536
min	32.100000	13.100000	172.000000	2700.000000
25%	39.225000	15.600000	190.000000	3550.000000
50%	44.450000	17.300000	197.000000	4050.000000
75%	48.500000	18.700000	213.000000	4750.000000
max	59.600000	21.500000	231.000000	6300.000000

Figure 14: Dataset statistics

The functions describe() provides a list of statical values such as: count, mean, std, min, max, and percentiles.

2.2.3.1 Central tendency

This part is concerned with mean, median, and mode.

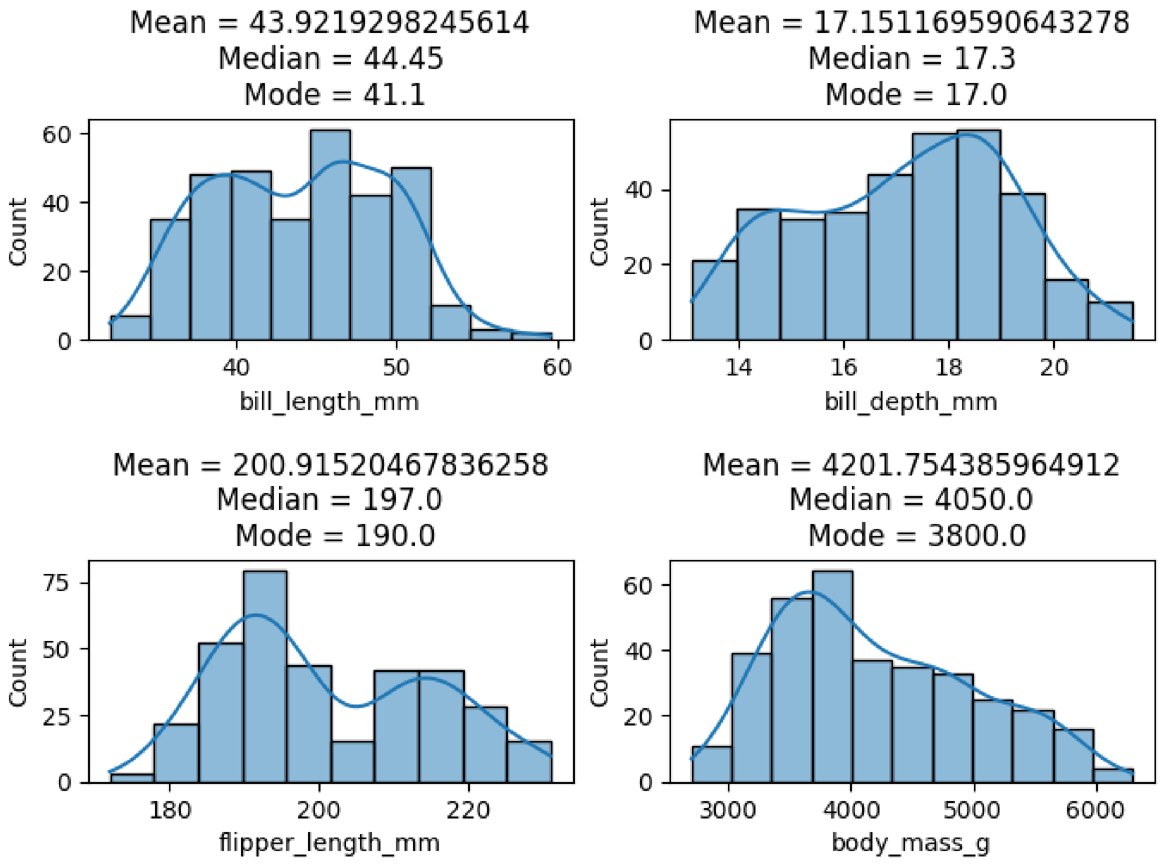


Figure 15: The distribution of features with their central tendency

The analysis of plots above:

- **Bill length**: it is close to a summation of two normal distributions, but the mode is slightly far from mean & median.
- **Bill depth**: Same as bill length, but the three measures are very close to each other, it could be a gaussian distribution.
- **Flipper length**: from the measures, it is not likely to be a normal distribution.
- **Body mass**: same as flipper length.

With these measures (**mean, median, mode**), it is unconfirmable that the distributions are normal (gaussian), we should use other measures.

2.2.3.2 Variation

This part focuses on the variance and the standard deviation.

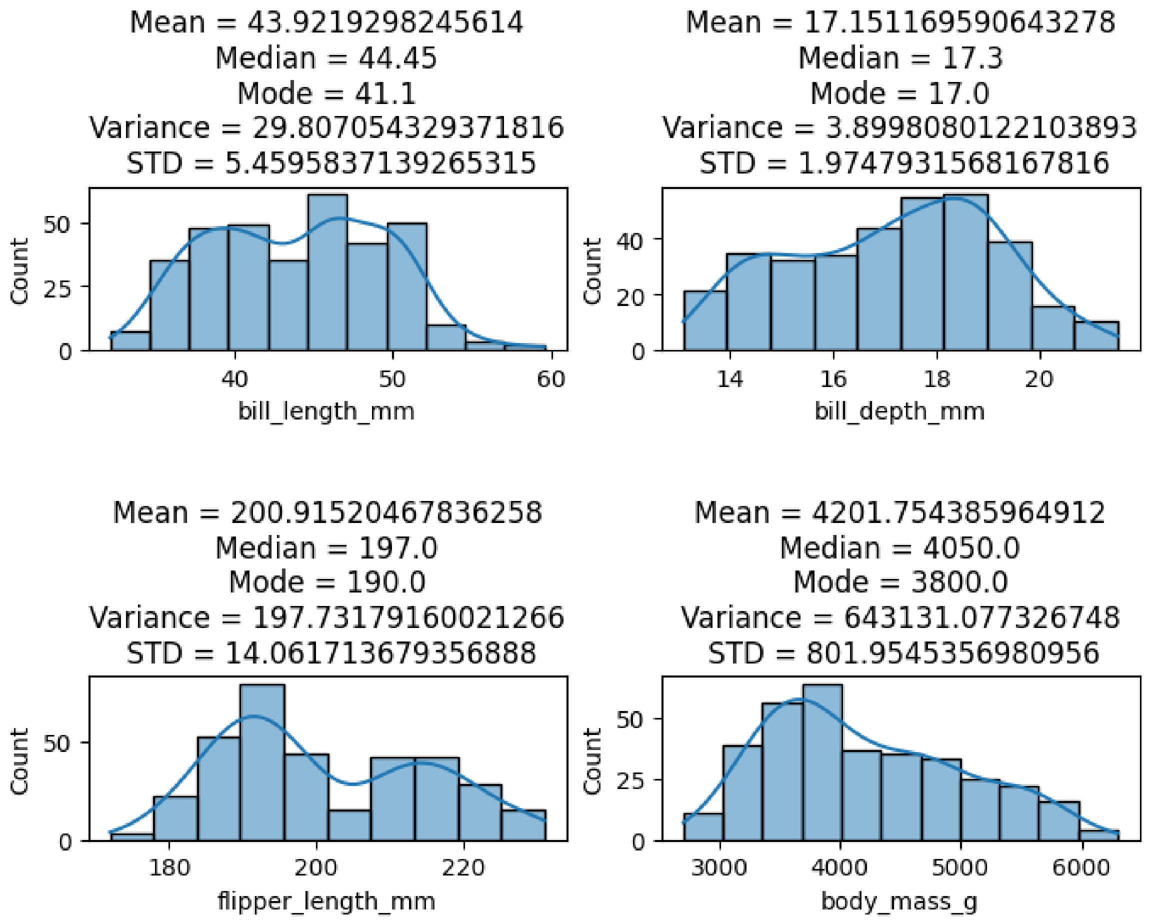


Figure 16: The distribution of features with their variation

The analysis of plots above:

1. **Bill length**: it has a moderate variance and standard deviation, suggesting moderate spread around the mean.
2. **Bill depth**: it has a low variance and standard deviation, indicating that the values are clustered closely around the mean.
3. **Flipper length**: it has a higher variance and standard deviation compared to the previous variables, indicating more spread around the mean.
4. **Body mass**: it has a significantly higher variance and standard deviation, suggesting considerable variability or dispersion around the mean, possibly due to outliers or a non-normal distribution.

Overall, these statistical values provide insights into the central tendency, spread, and symmetry of the distributions for each variable, helping to understand their characteristics and potential patterns in the data.

2.2.3.3 Shape of distribution

This part is about skewness and kurtosis.

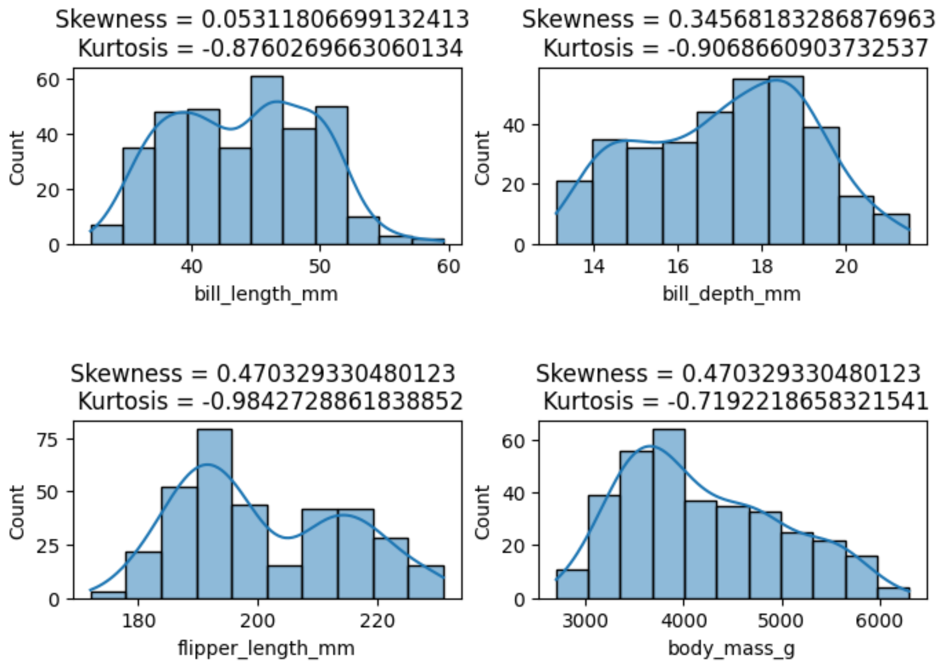


Figure 17: The distribution of features with their skewness and kurtosis

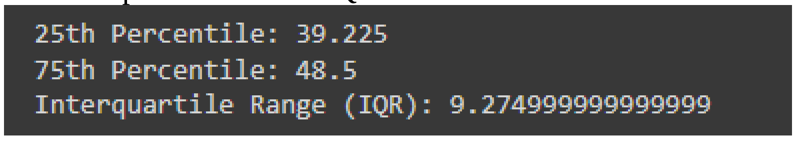
The analysis of plots above:

- All variables have negative kurtosis values, suggesting that they have lighter tails than a normal distribution.
- **Bill length** is nearly symmetric with a skewness close to 0.
- **Bill depth** is slightly negatively skewed.
- **Flipper length and Body mass** are positively skewed.

These values collectively describe the shape and characteristics of the distributions for each variable. Negative kurtosis indicates relatively less extreme values compared to a normal distribution, while skewness helps understand the asymmetry in the data distribution. These values can assist in identifying deviations from normality and understanding the nature of the data distributions.

2.2.3.4 Quantiles

This part focuses on the percentiles and IQR.



```
25th Percentile: 39.225
75th Percentile: 48.5
Interquartile Range (IQR): 9.274999999999999
```

Figure 18: Results of quantiles

- The 25th percentile (Q1) of 39.225 indicates that a quarter of the dataset's values lie below this point.
- The 75th percentile (Q3) at 48.5 shows that three-quarters of the dataset's values are less than or equal to this value.
- The IQR of approximately 9.275 suggests that the middle 50% of the data falls within this range, showing the spread of the central data points.

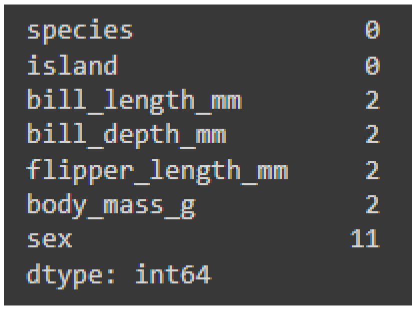
In summary, these quartile values and the IQR help understand the spread and dispersion of the central portion of the dataset, offering insights into its variability without being influenced by extreme values (outliers).

2.3 Data quality issues

This part in about detecting missing values and handling them, as well as detecting outliers and handling them if existing.

2.3.1 Detecting missing values

In the section, detecting missing values is done by find numerical values or displaying the rows that contains them.



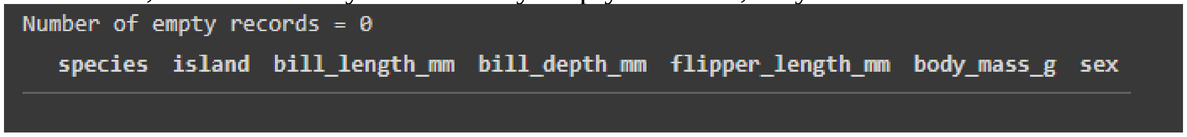
```
species               0
island                0
bill_length_mm        2
bill_depth_mm         2
flipper_length_mm     2
body_mass_g           2
sex                  11
dtype: int64
```

Figure 19: Features with the number of missing values

There are some missing values in the dataset.

- **species & island**: has no missing values.
- **bill_length_mm, bill_depth_mm, flipper_length_mm, body_mass_g**: each feature has 2 missing values.
- **sex**: has 11 missing values.

Sometimes, the dataset may contain fully empty rows. So, they must be addressed.

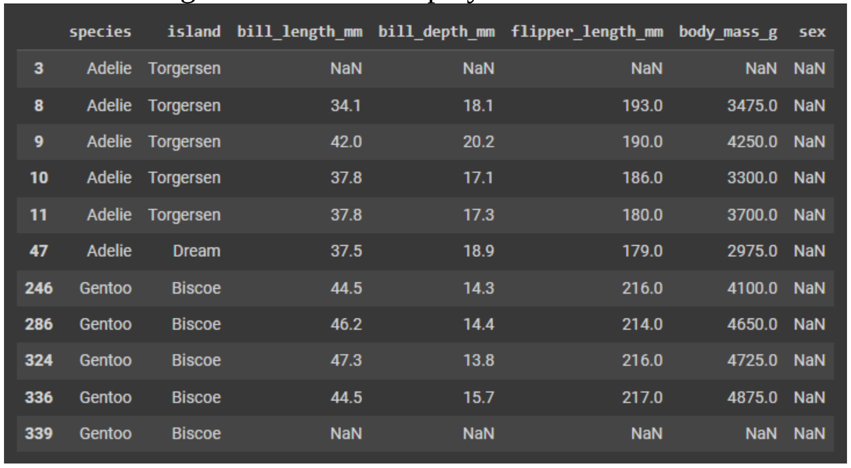


```
Number of empty records = 0
  species  island  bill_length_mm  bill_depth_mm  flipper_length_mm  body_mass_g  sex
```

Figure 20: Addressing fully empty rows

The dataset does not contain any row of this type.

Now, the rows with missing values will be displayed.



	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN
8	Adelie	Torgersen	34.1	18.1	193.0	3475.0	NaN
9	Adelie	Torgersen	42.0	20.2	190.0	4250.0	NaN
10	Adelie	Torgersen	37.8	17.1	186.0	3300.0	NaN
11	Adelie	Torgersen	37.8	17.3	180.0	3700.0	NaN
47	Adelie	Dream	37.5	18.9	179.0	2975.0	NaN
246	Gentoo	Biscoe	44.5	14.3	216.0	4100.0	NaN
286	Gentoo	Biscoe	46.2	14.4	214.0	4650.0	NaN
324	Gentoo	Biscoe	47.3	13.8	216.0	4725.0	NaN
336	Gentoo	Biscoe	44.5	15.7	217.0	4875.0	NaN
339	Gentoo	Biscoe	NaN	NaN	NaN	NaN	NaN

Figure 21: Dataset rows with missing values

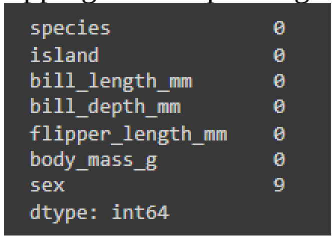
It's noticeable that there are two rows that have 2 values filled out from 7.

2.3.2 Handling missing values

2.3.2.1 Handling missing data through dropping

There are two samples filled with two features only out of 7. The percentage of used features is (28.57%). The proportion of samples is (0.58%). The proportion of the samples is very small and unnoticeable, so dropping them will not make any problem.

The approach of handling these missing values is by finding the percentage of null values in each row (sample). If this percentage exceeds the defined threshold (0.7, 70%), then the row will be dropped. Or, simply by dropping them depending on the proportion of the samples.



```
species              0
island               0
bill_length_mm       0
bill_depth_mm        0
flipper_length_mm    0
body_mass_g          0
sex                  9
dtype: int64
```

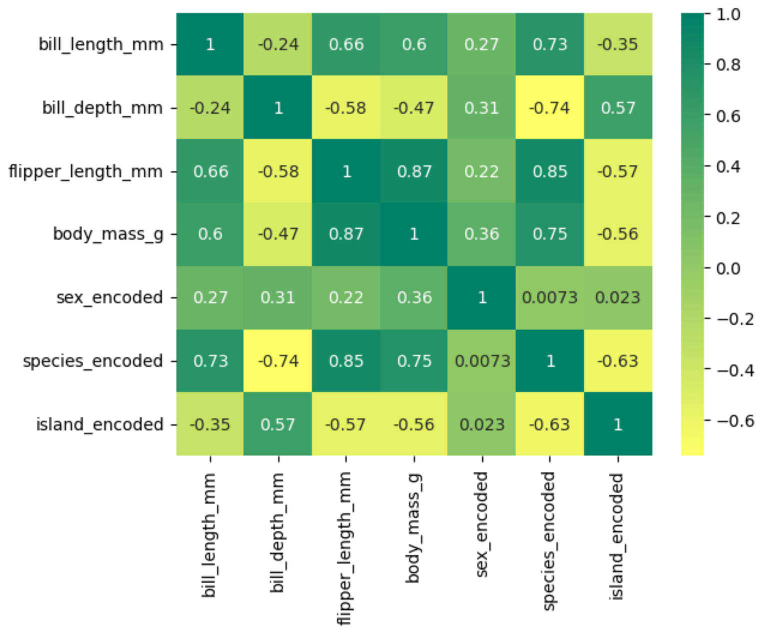
Figure 22: Features with the number of missing values

After handling the missing values through dropping, the targeted features were ('bill_length_mm', 'bill_depth_mm', 'flipper_length_mm', 'body_mass_g'). However, the figure above shows that the missing values were handled successfully. But, sex feature still not handled, but it will in the next section.

2.3.2.2 Handling missing data through imputation

The targeted feature in this part is sex.

To handle the sex feature, the highly-correlated feature with sex must be found to be used in imputing it. So, the heat map (confusion matrix) of features correlation must be displayed.



	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex_encoded	species_encoded	island_encoded
bill_length_mm	1	-0.24	0.66	0.6	0.27	0.73	-0.35
bill_depth_mm	-0.24	1	-0.58	-0.47	0.31	-0.74	0.57
flipper_length_mm	0.66	-0.58	1	0.87	0.22	0.85	-0.57
body_mass_g	0.6	-0.47	0.87	1	0.36	0.75	-0.56
sex_encoded	0.27	0.31	0.22	0.36	1	0.0073	0.023
species_encoded	0.73	-0.74	0.85	0.75	0.0073	1	-0.63
island_encoded	-0.35	0.57	-0.57	-0.56	0.023	-0.63	1

Figure 23: Correlation heatmap

Sex feature is highly correlated with body_mass_g feature, so it will be used in imputing the missing values.

However, the body_mass_g feature must be analyzed to choose the most suitable imputation technique.

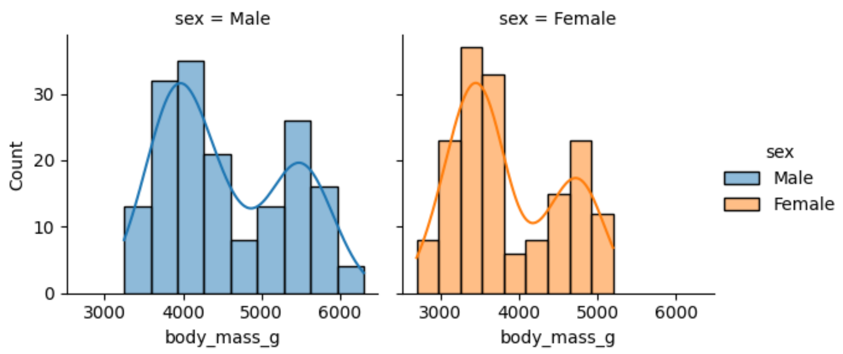


Figure 24: The distribution of body_mass_g with respect to sex

From the graph, both distributions look like the summation of two normal distributions. But statistical measure will give more accurate insights.

Table 1: body_mass_g statistics based on sex

Mean: Female 3862.272727 Male 4545.684524	Median: Female 3650.0 Male 4300.0	Mode: Female 3700.0 Male [3900.0, 3950.0]
STD: Female 666.172050 Male 787.628884	Skewness: Female 0.445063 Male 0.376064	Kurtosis: Female -1.119924 Male -1.204163

Based on the provided statistics:

- **Mean vs. Median vs. Mode**: For both genders (Female and Male), the mean, median, and mode values are not too different from each other, suggesting a potential tendency towards normality.
- **Skewness and Kurtosis**: The skewness and kurtosis values for both genders are close to 0. Skewness values around 0 (less than ±0.5) indicate relatively symmetric distributions, and kurtosis values around -1 to +1 indicate near-normal or mesokurtic distributions.

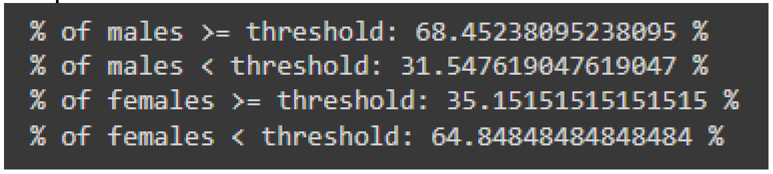
Given these statistics, both genders show characteristics that suggest a tendency toward normality in body mass distribution.

So, the choice between mean & median; in this case the mean & median values for both genders are relatively close to each other. Until now, things about outliers are not clear. So, using the median is better since both mean and median are close, but median is not sensitive to outliers.

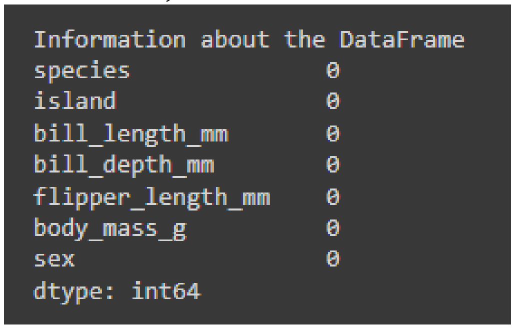
Imputation: the missing values will be imputed according to the **median** value. However, the threshold is the summation of the male and female body_mass_g medians divided by 2.

thresh = (f_median + m_median)/2

Since there are two values of the median (male and female), it is better to use their average is a threshold for imputation.


```
% of males >= threshold: 68.45238095238095 %
% of males < threshold: 31.547619047619047 %
% of females >= threshold: 35.15151515151515 %
% of females < threshold: 64.84848484848484 %
```
Figure 25: Males and females' percentages according to the defined threshold

The percentages mentioned above were used to decide how to use the defined threshold in imputing the missing values of sex feature. But, It is obvious that most males are greater than the body mass threshold, where most females are lesser than. So, any value greater than or equal this threshold is considered male, otherwise it is a female.



```
Information about the DataFrame
species              0
island               0
bill_length_mm       0
bill_depth_mm        0
flipper_length_mm    0
body_mass_g          0
sex                  0
dtype: int64
```

Figure 26: Features with the number of missing values

Now, the process of handling missing values is done since all features got **0** missing value.

2.3.3 Detecting outliers

This part is concerned about detecting outliers, there are various methods for detecting them. To summarize this part, the boxplots of targeted features displayed below:

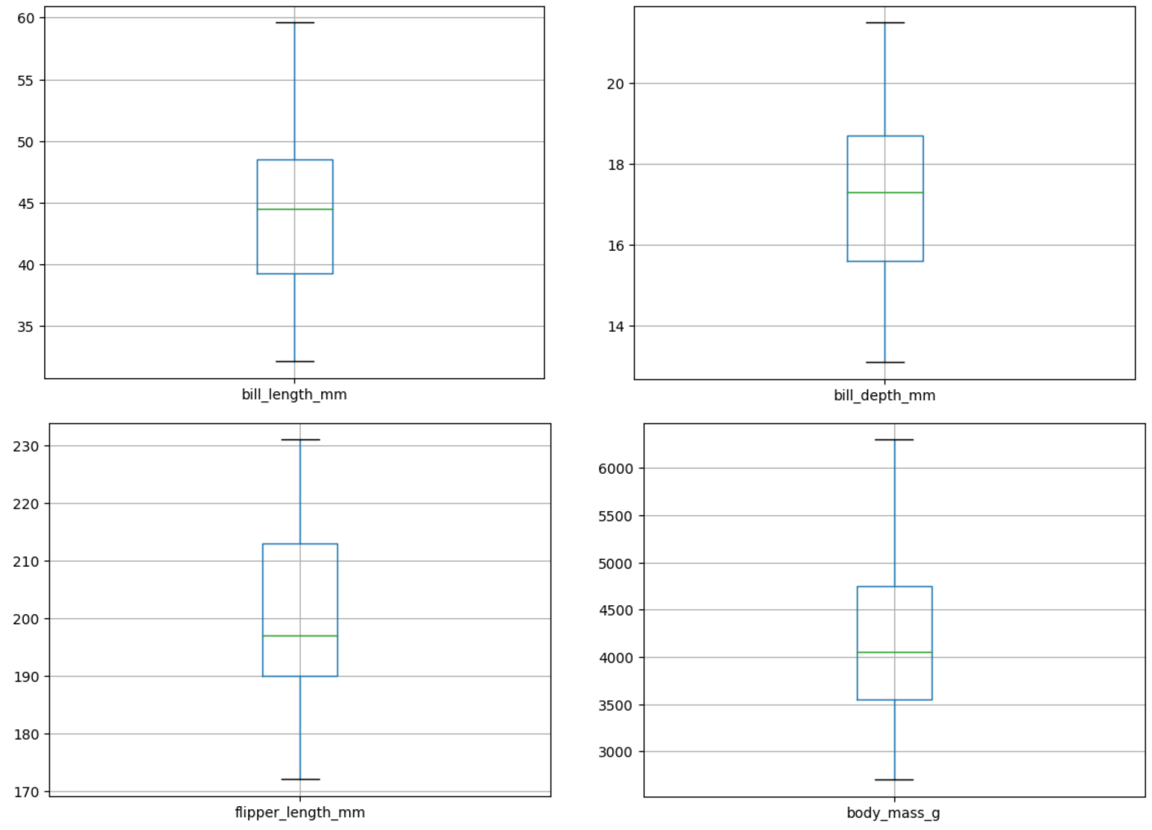


Figure 27: Detecting outliers using boxplots

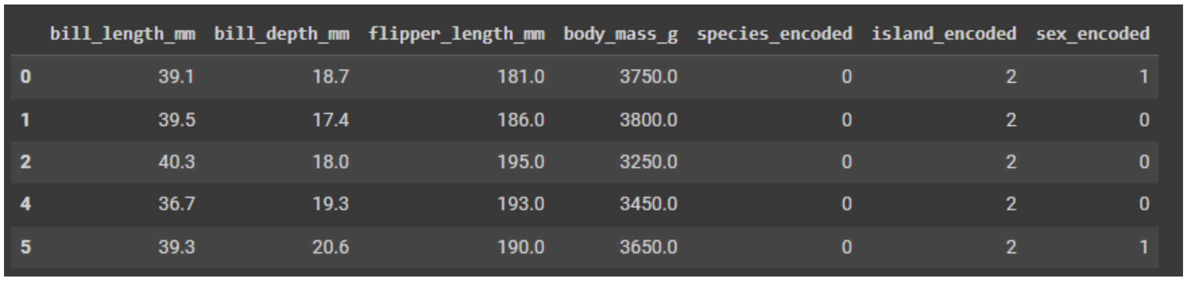
From the boxplots, the features do not have any outlier. There are different methods presented in the code to provide multiple ways of thinking in the analysis of outliers. But, to shorten it, boxplots were shown only. However, outliers detecting is applicable on numerical features only.

In the case study, feature selection is before the encoding process. But feature selection requires all feature to be numerical. So, I encoded the non-numerical features before feature selection.

2.4 Encoding categorical variables

In this case study, label encoder is used because it does not increase the number of features, as well as it keeps the original distribution. However, one-hot encoding was tested but didn't give prominent results.

There are three categorical features: sex, island, species. All of them are encoded using label encoder.



	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	species_encoded	island_encoded	sex_encoded
0	39.1	18.7	181.0	3750.0	0	2	1
1	39.5	17.4	186.0	3800.0	0	2	0
2	40.3	18.0	195.0	3250.0	0	2	0
4	36.7	19.3	193.0	3450.0	0	2	0
5	39.3	20.6	190.0	3650.0	0	2	1

Figure 28: Result of encoding process

The figure above is a snippet of the dataset after encoding the three features.

2.5 Split the data into X, y

This step is important for all upcoming parts. For example, feature selection can't be done without splitting the data into X, y

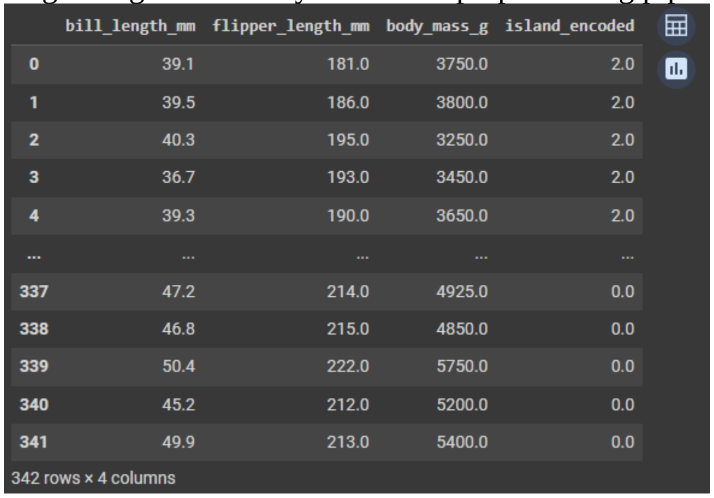
We split the data into input [X] and output [y], the target feature is the **species** of the penguin. It is possible to use other features, but after testing other features, the best accuracy is obtained when using species as the target feature.

2.6 Feature selection

In this part, there are two main parameters required for feature selection:

- Filtering method
- K: number of top features to select.

For both parameters, the best filtering method is chi2 (chi-squared test). And, the value of k is 4. These values were obtained after repeating the case study multiple times (1000). The selected features using chi2 gave accuracy 1 after the preprocessing pipeline.

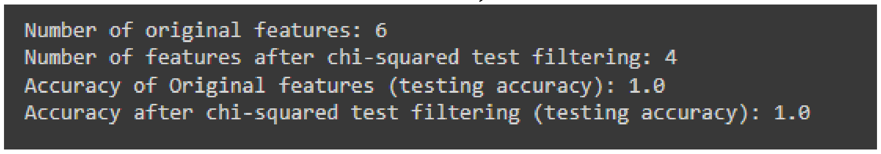


	bill_length_mm	flipper_length_mm	body_mass_g	island_encoded
0	39.1	181.0	3750.0	2.0
1	39.5	186.0	3800.0	2.0
2	40.3	195.0	3250.0	2.0
3	36.7	193.0	3450.0	2.0
4	39.3	190.0	3650.0	2.0
...	...	...	...	...
337	47.2	214.0	4925.0	0.0
338	46.8	215.0	4850.0	0.0
339	50.4	222.0	5750.0	0.0
340	45.2	212.0	5200.0	0.0
341	49.9	213.0	5400.0	0.0

342 rows × 4 columns

Figure 29: Selected features

Filtered features are sex_encoded & bill_depth_mm.
To ensure the correctness of the feature selection, the selected feature must be evaluated.



```
Number of original features: 6
Number of features after chi-squared test filtering: 4
Accuracy of Original features (testing accuracy): 1.0
Accuracy after chi-squared test filtering (testing accuracy): 1.0
```

Figure 30: Evaluation of feature selection

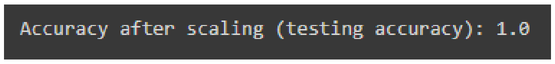
The accuracy is 1.0 after dropping the two features. However, this is the best-case solution because I tried four methods with different parameters. Moreover, the number of features is 4.

2.7 Split the dataset into training and testing subsets

This part is very simple, after splitting the data into X, y and filtering the features of X. After that, the filtered X and y are split into training and testing subsets.

2.8 Scale or normalize the numerical features

This section is about scaling the numerical features. The dataset is split in the previous step. However, the scaling targets both training and testing X subsets. Moreover, there are two techniques for scaling: min-max and standardization. Both ways were tested along the case study, standardization gave prominent results and kept the value of the accuracy. This is due to the reshaping of original features.



```
Accuracy after scaling (testing accuracy): 1.0
```

Figure 31: The accuracy after scaling

There is a visualized example on the features after scaling.

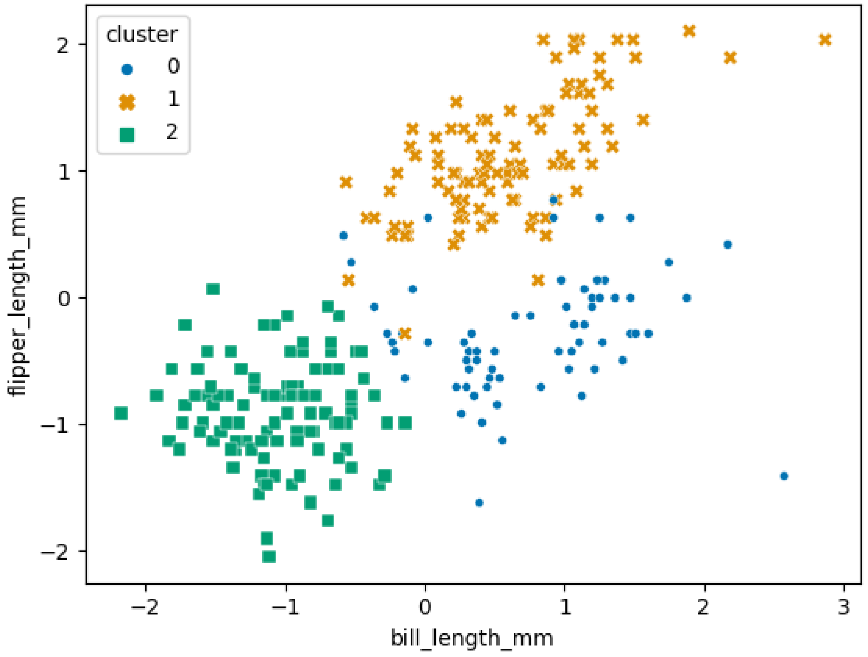


Figure 32: The relation between bill_length_mm and flipper_length_mm

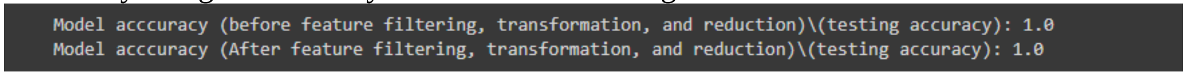
After scaling, the relationship between both feature is clear. Each group of points is contained in one cluster.

2.9 Apply suitable dimensionality reduction techniques

This part is the last step in preprocessing the dataset. After feature selection and scaling, now it is the time to reduce the dimensionality of the dataset. In other words, reducing the number of features. PCA is used for this part, and there is a main parameter for the PCA, which is n_components (number of components to keep). After many tests, the best value is 3.

2.10 Validate your preprocessing pipeline

The validation of the preprocessing pipeline (feature filtering, transformation, and reduction) is done by using the accuracy measure on the testing set.



```
Model acccuracy (before feature filtering, transformation, and reduction)\(testing accuracy): 1.0
Model acccuracy (After feature filtering, transformation, and reduction)\(testing accuracy): 1.0
```

Figure 33: Models accuracy before and after preprocessing

It is clear the preprocessing pipeline is good since there are no loss in the accuracy. However, this process is done after many tests in choosing the best values for parameters along with the most suitable techniques for each part.

3 Conclusion

The analysis of the Penguins dataset underscores the critical role of data preprocessing in preparing datasets for machine learning. Visualizations employing various libraries offer intuitive insights into the dataset, aiding in pattern recognition and understanding. Descriptive statistics provide an initial overview, crucial for identifying patterns, anomalies, and potential issues affecting model quality. Handling missing data involves judicious decisions depending on the amount and nature of missingness. Similarly, outlier detection and handling necessitate careful considerations, ensuring the balance between data integrity and cleaning. Feature selection methods contribute to improving model performance by reducing dimensionality and eliminating irrelevant features. Finally, data transformation techniques ensure the data is formatted suitably for machine learning models. This case study underscores the significance of these preprocessing steps in enhancing the quality and reliability of subsequent machine learning analysis.