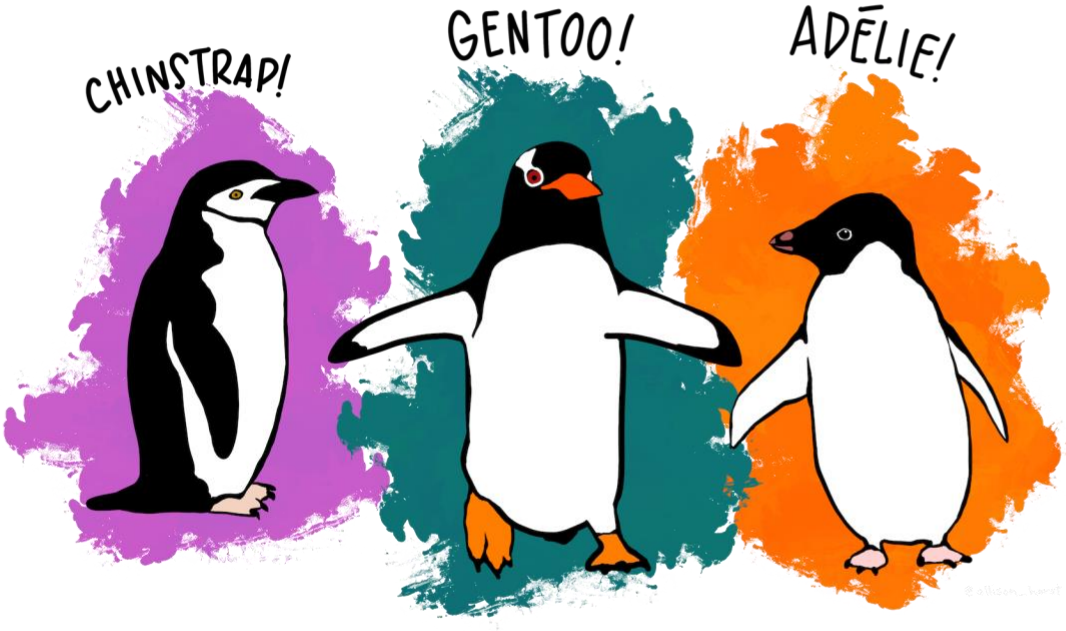
Final Assignment (Palmer Penguins Dataset)

Report, JUNE 2024



Data Mining

Antonio Brimes Romero

INDEX

1. Introduction

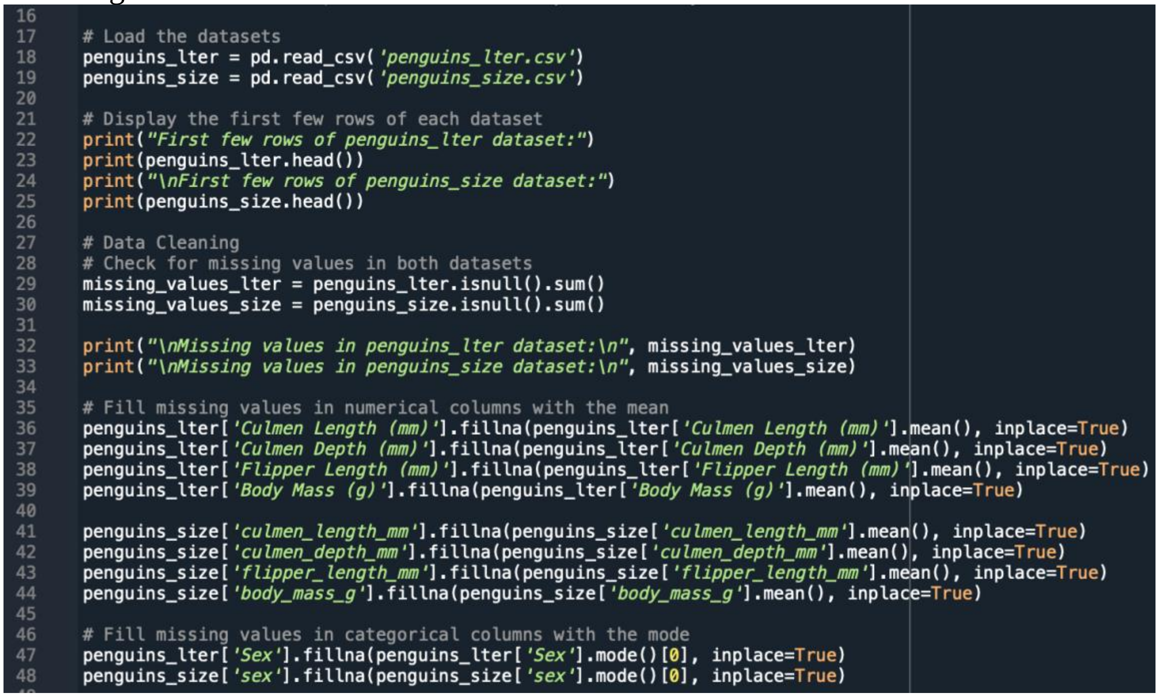
The Palmer Penguins dataset includes size measurements for three penguin species found on three islands in the Palmer Archipelago, Antarctica. The goal of this investigation is to do an explanatory data analysis (EDA) on the dataset to get an understanding of the relationships between the penguins' various physical parameters. We want to answer the following questions:

1. What are the main characteristics and distributions of measurements in the dataset?
2. Are there any significant correlations between these parameters and the penguin species?

2. Methodology and results

2.1. Data cleaning

We started by importing the 'penguins_lter' and 'penguins_size' datasets. Then, we look for missing data and processing them accordingly, filling numerical columns with the mean and categorical columns with the mode.

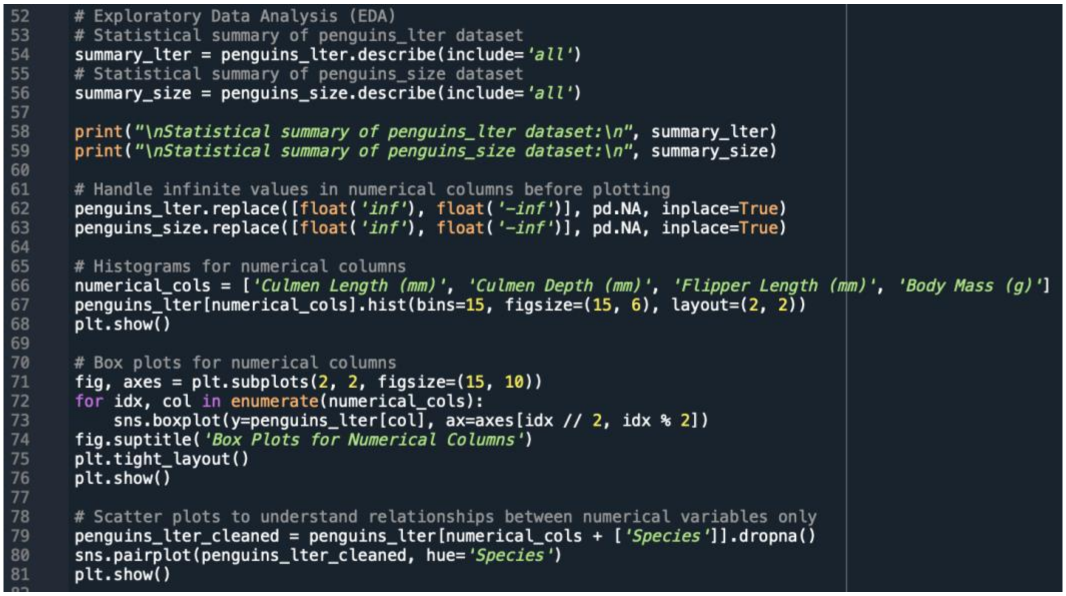


```python

# Load the datasets
penguins_lter = pd.read_csv('penguins_lter.csv')
penguins_size = pd.read_csv('penguins_size.csv')

# Display the first few rows of each dataset
print("First few rows of penguins_lter dataset:")
print(penguins_lter.head())
print("\nFirst few rows of penguins_size dataset:")
print(penguins_size.head())

# Data Cleaning
# Check for missing values in both datasets
missing_values_lter = penguins_lter.isnull().sum()
missing_values_size = penguins_size.isnull().sum()

print("\nMissing values in penguins_lter dataset:\n", missing_values_lter)
print("\nMissing values in penguins_size dataset:\n", missing_values_size)

# Fill missing values in numerical columns with the mean
penguins_lter['Culmen Length (mm)'].fillna(penguins_lter['Culmen Length (mm)'].mean(), inplace=True)
penguins_lter['Culmen Depth (mm)'].fillna(penguins_lter['Culmen Depth (mm)'].mean(), inplace=True)
penguins_lter['Flipper Length (mm)'].fillna(penguins_lter['Flipper Length (mm)'].mean(), inplace=True)
penguins_lter['Body Mass (g)'].fillna(penguins_lter['Body Mass (g)'].mean(), inplace=True)

penguins_size['culmen_length_mm'].fillna(penguins_size['culmen_length_mm'].mean(), inplace=True)
penguins_size['culmen_depth_mm'].fillna(penguins_size['culmen_depth_mm'].mean(), inplace=True)
penguins_size['flipper_length_mm'].fillna(penguins_size['flipper_length_mm'].mean(), inplace=True)
penguins_size['body_mass_g'].fillna(penguins_size['body_mass_g'].mean(), inplace=True)

# Fill missing values in categorical columns with the mode
penguins_lter['Sex'].fillna(penguins_lter['Sex'].mode()[0], inplace=True)
penguins_size['sex'].fillna(penguins_size['sex'].mode()[0], inplace=True)
```

By using this code, we ensure that the datasets are complete and ready for analysis by addressing missing values. Numerical columns are filled with the mean value, while categorical columns are filled with the mode value.

- Exploratory Data Analysis (EDA)

  We used statistical measures like mean, median, and standard deviation to summarize the dataset's main features. We used histograms, box plots, and pair plots to understand the data distribution and variable relationships.



```python
# Exploratory Data Analysis (EDA)
# Statistical summary of penguins_lter dataset
summary_lter = penguins_lter.describe(include='all')
# Statistical summary of penguins_size dataset
summary_size = penguins_size.describe(include='all')

print("\nStatistical summary of penguins_lter dataset:\n", summary_lter)
print("\nStatistical summary of penguins_size dataset:\n", summary_size)

# Handle infinite values in numerical columns before plotting
penguins_lter.replace([float('inf'), float('-inf')], pd.NA, inplace=True)
penguins_size.replace([float('inf'), float('-inf')], pd.NA, inplace=True)

# Histograms for numerical columns
numerical_cols = ['Culmen Length (mm)', 'Culmen Depth (mm)', 'Flipper Length (mm)', 'Body Mass (g)']
penguins_lter[numerical_cols].hist(bins=15, figsize=(15, 6), layout=(2, 2))
plt.show()

# Box plots for numerical columns
fig, axes = plt.subplots(2, 2, figsize=(15, 10))
for idx, col in enumerate(numerical_cols):
    sns.boxplot(y=penguins_lter[col], ax=axes[idx // 2, idx % 2])
fig.suptitle('Box Plots for Numerical Columns')
plt.tight_layout()
plt.show()

# Scatter plots to understand relationships between numerical variables only
penguins_lter_cleaned = penguins_lter[numerical_cols + ['Species']].dropna()
sns.pairplot(penguins_lter_cleaned, hue='Species')
plt.show()
```

The statistics summaries offer a comprehensive overview of the information. Measurement distribution is represented by histograms (figure 1), spread and possible outliers are shown by box plots (figure 2), and correlations between measurement pairs are displayed by pair plots (figure 3), we obtain the following output:

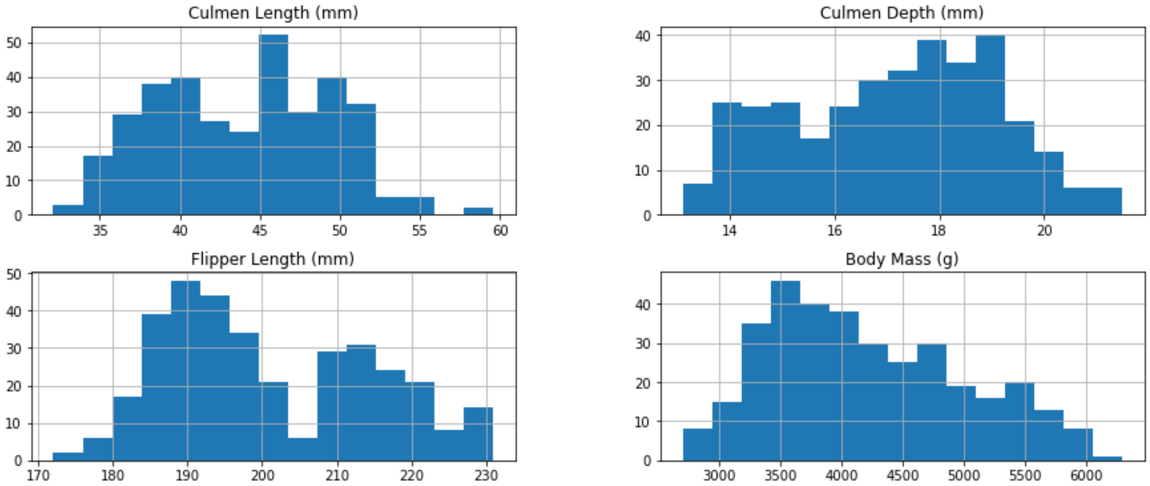


Fig 1. Histogram

The histograms show the distribution of culmen length, culmen depth, flipper length, and body mass across the dataset.

- Culmen Length: Most values range between 35 mm and 50 mm.
- Culmen Depth: Values are mostly between 14 mm and 20 mm.
- Flipper Length: Ranges from 170 mm to 230 mm with a peak around 190 mm.
- Body Mass: Majority of the values lie between 3000 g and 5000 g.

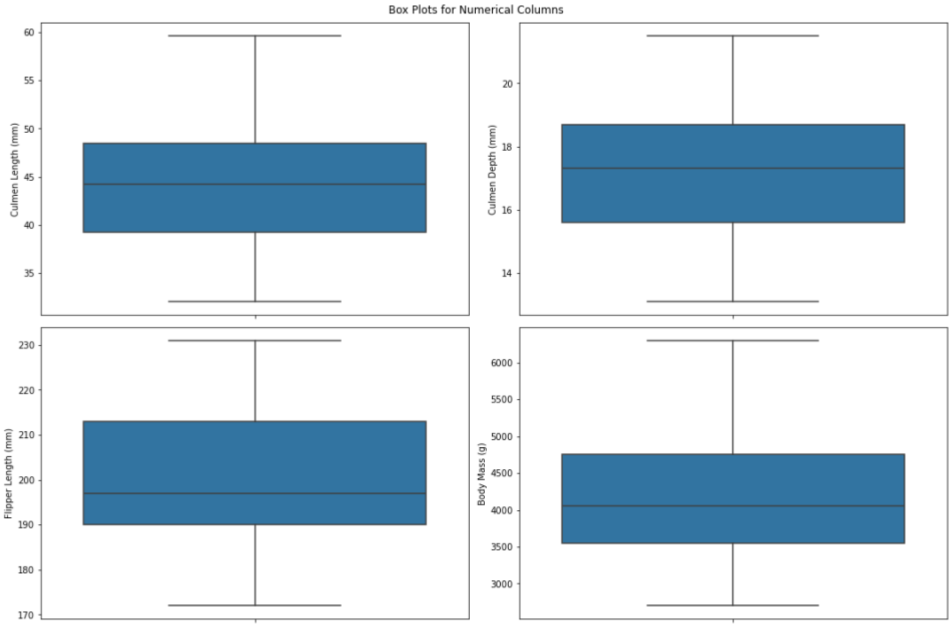


Fig 2. Boxplot

The box plots provide a summary of the spread and central tendencies of the numerical variables.

- Culmen Length: Median around 45 mm, with a few outliers above 50 mm.
- Culmen Depth: Median around 18 mm, with a uniform spread.
- Flipper Length: Median around 200 mm, showing some outliers above 220 mm.
- Body Mass: Median around 4000 g, with some high outliers above 5500 g.

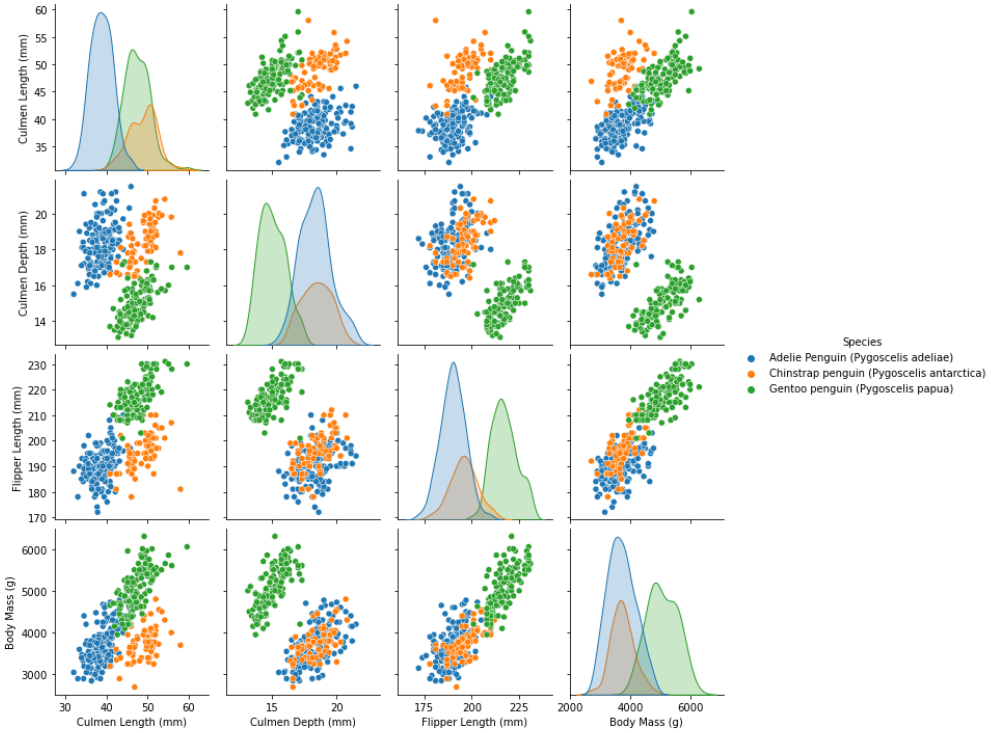


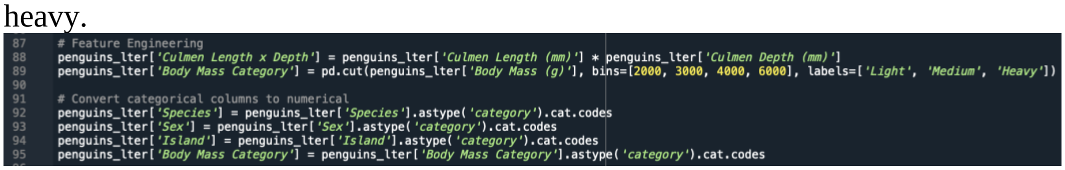
Fig 3. Pair plot

Pair plots show the relationships between all pairs of numerical variables.

- There are strong positive correlations between flipper length and body mass, and between culmen length and flipper length.
- Species clustering can be observed in these pairwise plots.

- Feature Engineering

  We added new features based on the initial measurements to improve the dataset's predictive capacity. For instance, we included a feature for the product of culmen length and depth and divided body mass into three categories: light, medium, and heavy.

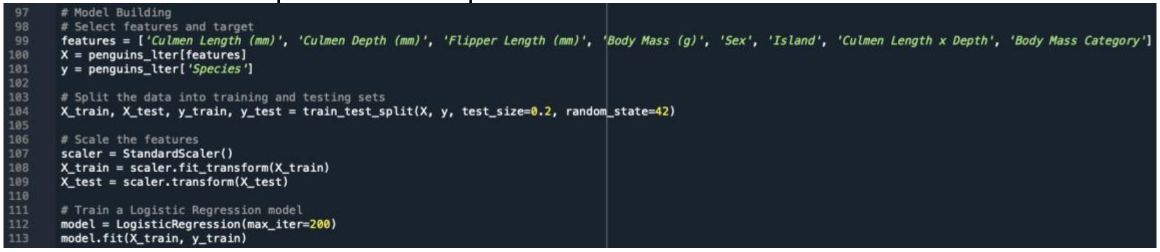


```
# Feature Engineering
penguins_lter['Culmen Length x Depth'] = penguins_lter['Culmen Length (mm)'] * penguins_lter['Culmen Depth (mm)']
penguins_lter['Body Mass Category'] = pd.cut(penguins_lter['Body Mass (g)'], bins=[2000, 3000, 4000, 6000], labels=['Light', 'Medium', 'Heavy'])

# Convert categorical columns to numerical
penguins_lter['Species'] = penguins_lter['Species'].astype('category').cat.codes
penguins_lter['Sex'] = penguins_lter['Sex'].astype('category').cat.codes
penguins_lter['Island'] = penguins_lter['Island'].astype('category').cat.codes
penguins_lter['Body Mass Category'] = penguins_lter['Body Mass Category'].astype('category').cat.codes
```

This step includes additional features and transform categorical variables into numerical representation, preparing the data for model training.

2.2. Model building

We chose logistic regression because it is simple and successful in classification tasks. The data was divided into training and testing sets, and features were adjusted with ‘StandardScaler’ to optimize model performance.

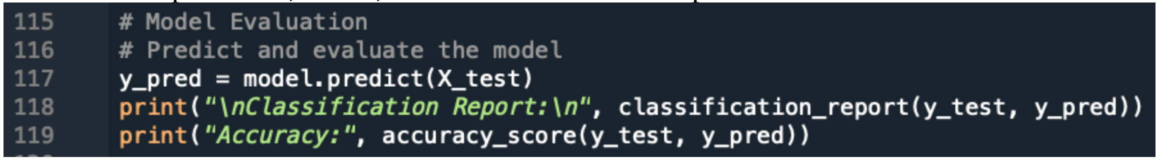


```
# Model Building
# Select features and target
features = ['Culmen Length (mm)', 'Culmen Depth (mm)', 'Flipper Length (mm)', 'Body Mass (g)', 'Sex', 'Island', 'Culmen Length x Depth', 'Body Mass Category']
X = penguins_lter[features]
y = penguins_lter['Species']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Scale the features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Train a Logistic Regression model
model = LogisticRegression(max_iter=200)
model.fit(X_train, y_train)
```

The logistic regression model was trained on the selected features, and the data was divided to allow the model to be tested on previously unknown data.

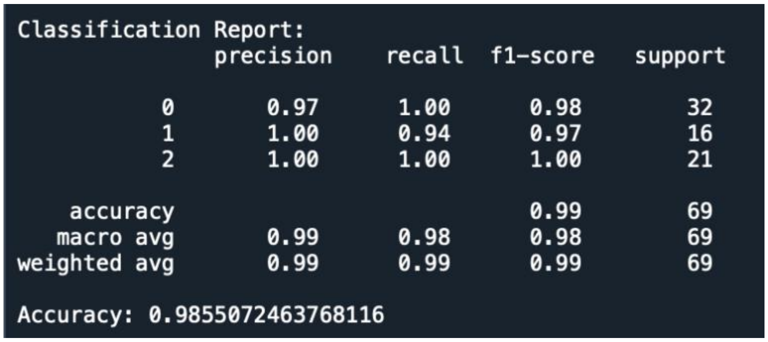
2.3. Model evaluation

The model's performance was evaluated using accuracy and a classification report to calculate the precision, recall, and F1-score for each species class.



```
# Model Evaluation
# Predict and evaluate the model
y_pred = model.predict(X_test)
print("\nClassification Report:\n", classification_report(y_test, y_pred))
print("Accuracy:", accuracy_score(y_test, y_pred))
```

The model obtained an accuracy of around 98.5%, showing strongly accuracy for identifying penguin species using the selected features, the result of one running is the following:



```
Classification Report:
              precision    recall  f1-score   support

           0       0.97      1.00      0.98        32
           1       1.00      0.94      0.97        16
           2       1.00      1.00      1.00        21

    accuracy                           0.99        69
   macro avg       0.99      0.98      0.98        69
weighted avg       0.99      0.99      0.99        69

Accuracy: 0.9855072463768116
```

3. Interpretation

The logistic regression model accurately identified penguin species based on the selected features. Key characteristics such as flipper length, culmen length, and body mass were strongly correlated with species classification. The implementation of new features such as 'Culmen Length x Depth' and body mass classification also helped to improve the model's efficacy.

- Key findings:
  - Flipper length is one of the most important predictors of species.
  - Culmen Measurements: The length and depth of the culmen are important in distinguishing species.
  - Body Mass: Increases predictive power, especially when categorized.

4. Conclusion

In our analysis of the Palmer Penguins dataset, we tried to classify penguin species using logistic regression. We started by cleaning the data to remove missing values, followed by exploratory data analysis (EDA) to summarize significant characteristics and correlations in the data. To increase model performance, feature engineering comprised the creation of interaction variables and the classification of body mass.

Our logistic regression model had an accuracy of 98.5%, showing a great ability to detect penguin species using the available data. Despite its excellent accuracy, the method has limitations such as addressing missing values via imputation, a short sample size, and potential variability between features. These factors could affect the model's generalizability and stability.

Overall, our method successfully applied EDA and logistic regression to identify important insights and achieve high classification accuracy in the Palmer Penguin dataset. Addressing the identified errors could improve the result's accuracy.

5. References

Seaborn documentation that we use for the data visualization in which we can see examples of the usage of the library with the palmer penguin dataset: https://seaborn.pydata.org/