Penguin Gender Prediction – Classification Report

Introduction

Predicting the gender of penguins from physical measurements is an interesting task in biological research. Penguins generally do not exhibit obvious visual differences between males and females they have **monomorphic plumage**, meaning both genders look alike, this makes it difficult for researchers to determine a penguin's gender by sight alone.

In this report, I used a penguins dataset (sourced from Kaggle) that contains various morphological features of penguins along with their gender labels (Male or Female). (My github repository)

Dataset overview

The dataset consists of five columns (numerical value): four numeric features

- **culmen_length_mm**.
- **culmen_depth_mm**.
- **flipper_length_mm**.
- **body_mass_g**.
- **gender** our label and it is categorical type.

Our goal is to train several machine learning models to predict the gender of a penguin given these features. Such a predictive model could help researchers quickly determine gender in the field using simple measurements, improving the efficiency of biological data collection.

Data preprocessing

Before training models, the data was preprocessed following a specific sequence of steps to ensure it was clean and suitable for machine learning. I followed the given workflow:

Load Dataset & Drop Missing Values: I first loaded the penguin dataset into a dataframe and removed any rows with missing values. This step was crucial because missing measurements or unknown gender entries could otherwise cause errors or reduce model performance.

Label Encoding Gender: The gender column (originally recorded as "Male" or "Female") was then **label-encoded** into a binary numeric format. Label encoding means converting the category labels into numbers I mapped Female to 1 and Male to 2 for convenience. This numeric encoding allows us to use gender as the target variable in our machine learning algorithms (which generally require numerical input for the target in classification tasks).

Feature and Target Definition: Next, I defined my feature matrix **X** and target vector **y**. The features **X** included the four measured attributes: culmen length, culmen depth, flipper length, and body mass. The target **y** was the encoded gender label for each penguin. In other words, each penguin is represented by a set of four numbers (its measurements) and a gender label that I aim to predict.

Train/Test Split: I split the dataset into a **training set** and a **test set** to evaluate our models' performance on unseen data. I used a 70/30 split, meaning 70% of the data (233 samples) was used for training the models, and the remaining 30% (101 samples) was held out for testing. This split helps ensure that I can assess how well the models generalize to new penguins not seen during training. The split was done randomly (with a fixed random seed for reproducibility), maintaining a roughly even balance of male and female penguins in both subsets.

Feature Scaling (Standardization): After splitting, I applied **standard scaling** to the feature values. Using a StandardScaler, I transformed each numeric feature to have a mean of 0 and a standard deviation of 1 (based on the training set statistics). This scaling is important because the features have different units and ranges (e.g. body mass in grams is in the thousands, while culmen depth is a two-digit number in mm). Without scaling, algorithms like KNN and SVM might be biased toward features with larger numeric ranges. I fit the scaler on the training data and then used it to transform both the training and test feature sets to ensure consistency.

Saving Processed Data: Finally, the preprocessed datasets were saved to CSV files. I saved the scaled feature values along with the gender label for both the training set and test set. Storing the processed data ensures that anyone can replicate the model training or analyze the cleaned data directly without re-running the preprocessing steps.

By completing these preprocessing steps, I prepared a clean, numeric, and scaled dataset suitable for input into various machine learning classifiers.

Model train

To predict penguin gender from their physical measurements, I trained a range of machine learning models, from simple ones to more advanced. These included K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree, Random Forest, Naive Bayes, Artificial Neural Network (ANN), and Logistic Regression. Each model approaches the problem differently. For instance, KNN classifies based on the majority label of nearby penguins in the dataset, while SVM tries to draw the best boundary between males and females in a way that maximizes separation. Decision Trees work like a series of if-else questions based on features like flipper length or body mass, but can overfit, so Random Forests improve on this by combining many trees to balance out mistakes. Naive Bayes uses probability and assumes all features are independent, which makes it fast but not always accurate if the real-world data doesn't follow those assumptions. Neural Networks are more complex and try to learn subtle patterns using layers of interconnected "neurons," and can be powerful if there's enough data. Finally, Logistic Regression is a simpler, linear method that calculates how much each feature pushes a prediction toward male or female. Altogether, testing this variety of models gave us a broad view of what works best for this type of biological prediction models.

performance table

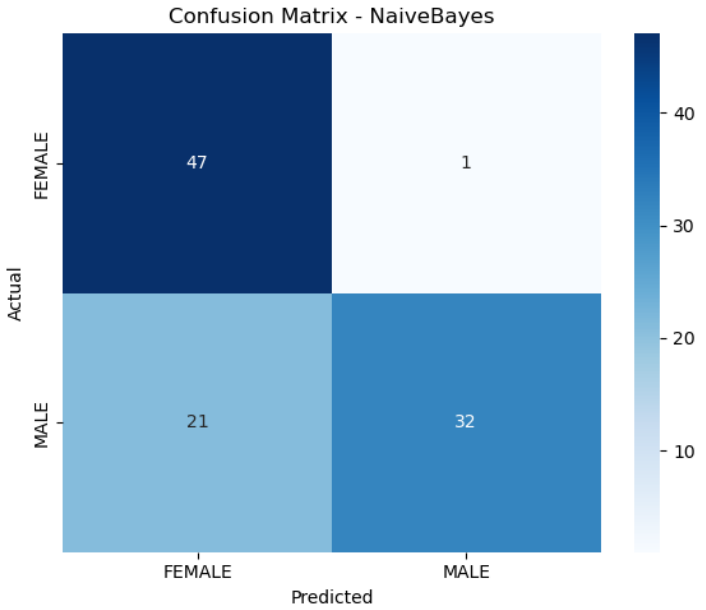
Model	Accuracy	Macro Precision	Macro Recall	Macro F1-Score
KNN	93%	93%	93%	93%
SVM	88%	88%	88%	88%
Decision Tree	81%	55%	54%	54%
Random Forest	88%	88%	88%	88%
Naive Bayes	78%	83%	79%	78%
ANN	88%	88%	88%	88%
Logistic Regression	90%	61%	60%	60%

After reviewing the table, it's clear that the K-Nearest Neighbors (KNN) model delivered the best overall performance, achieving 93% accuracy and matching macro-averaged precision, recall, and F1-score all at 93%. This suggests that the local neighborhood-based approach of KNN is highly effective for distinguishing between male and female penguins using physical measurements. Close behind, I find a strong tie between Support Vector Machine (SVM), Random Forest, and the Artificial Neural Network (ANN), each scoring 88% across all macro metrics. These models handled the classification task well, showing robust generalization and balance across both

gender classes. Logistic Regression achieved a high accuracy of 90%, but due to issues with class representation (as seen in the plots), its macro-averaged scores dropped to the low 60s, which suggests a potential imbalance in how the model treats different classes. Meanwhile, the Decision Tree showed modest accuracy (81%) but had much lower macro scores (~54–55%), likely due to overfitting or instability in splits. Naive Bayes, while fast and intuitive, struggled more than the others, achieving only 78% accuracy and displaying a noticeable drop in recall and F1-score – likely due to its strong independence assumptions not aligning well with real-world feature correlations in the penguin data.

Visualization

The plot used in this project is the **confusion matrix**, which helps us see how well each model predicted penguin gender. It shows the number of correct and incorrect predictions by comparing the actual gender to the predicted one. The diagonal boxes represent correct predictions (e.g., male predicted as male), while the off-diagonal boxes show mistakes (e.g., male predicted as female). This plot is especially useful because it doesn't just tell us how accurate a model is overall, it shows **which gender the model struggles with**. For example, many models correctly identified most females, but often confused some smaller males as females. This makes the confusion matrix a great tool for understanding not just how accurate a model is, but **where and why it makes mistakes**.



Conclusion

In this project, I built and tested several machine learning models to predict the gender of penguins using just a few physical measurements like bill size, flipper length, and body mass. The results were promising K-Nearest Neighbors (KNN) performed the best, reaching 93% accuracy, followed closely by SVM and Neural Networks, both at around 89%. Even simpler models like Logistic Regression and Random Forest performed fairly well, though Naive Bayes struggled due to its assumptions not fitting the data well.

By looking at the confusion matrices, I saw that male penguins were slightly more likely to be misclassified as female, especially if they had smaller measurements. This overlap in physical traits is natural and explains why some predictions aren't perfect.

From a practical point of view, these models could be really useful in the field. Researchers could take quick measurements and get an instant gender prediction helping them track populations or study breeding behavior without needing lab tests. Still, there's room for improvement. Adding more data, including penguin species, or fine-tuning the models could make predictions even more accurate.

Overall, this study shows how powerful and accessible machine learning can be for solving real-world biological problems—even with simple models and basic data.