Palmer Archipelago Penguin Dataset

Creation of K-Nearest Neighbors and Naive Bayes machine learning classification models, *Alessandro Sciorilli*

The following report presents the methodology and outcomes of employing the K-Nearest Neighbors and Naive Bayes machine learning classification models on the Palmer Archipelago Penguin Dataset sourced from GitHub. This dataset contains valuable information on penguin species (Adelie, Gentoo, and Chinstrap), along with attributes like bill length and depth, flipper length, body mass, and sex. For this study, I designated '*species*' as the label, while the remaining variables were considered as features in the analysis.

After importing the dataset into a Pandas Dataframe, I explored the data and checked for missing values. I found that 2 values were missing from the columns '*bill_length_mm'*, '*bill_depth_mm'*, '*flipper_length_mm*', and '*body_mass_g*', while a total of 11 values were missing from the '*sex*' column. I proceeded to remove the rows with missing values from the dataset.

As the next step in my analysis, I generated data visualizations to gain better insights into the distribution of values in my database, focusing on the three observed classes of penguins. In **Plot 2**, which shows the number of penguins per species, I observed that the most common species is Adelie with 146 registered specimens, followed by Gentoo with 119 specimens, and Chinstrap with only 68 specimens. Moving on to **Plot 3**, a scatterplot displaying the relationship between bill length and depth among the three species, distinct patterns emerged. Adelie penguins tend to have a shorter bill length and larger bill depth, while Gentoo penguins exhibit the opposite trend. On the other hand, Chinstrap penguins have both a longer bill length and larger bill depth. The clear separation among the species on the scatterplot indicates that bill dimensions play a crucial role in distinguishing the different penguin species. In **Plot 4**, a boxplot, it becomes evident that the Gentoo species stands out as having the largest body mass, while Adelie and Chinstrap species show similar lower body mass values. Regarding flipper length, **Plot 5**, a swarmplot, reveals that Gentoo penguins have the longest flippers, while Adelie and Chinstrap penguins report lower and similar values. Lastly, an interesting finding from **Plot 6** is that the Adelie species is present on all three islands of Antarctica (Torgersen, Biscoe, and Dream), while the Gentoo species can only be found on Biscoe Island, and the Chinstrap species exclusively inhabits Dream Island. This information sheds light on the distribution of these penguin species across the Antarctic islands.

To prepare the data for analysis, I removed the columns '*rowid*' and '*year*' since they were not relevant to my study. Next, I used **one-hot encoding** on the categorical variables '*island*' and '*sex*' to convert them into a numerical format suitable for machine learning analysis. One-hot encoding involves creating separate binary (0 or 1) dummy variables for each sub-category of the '*island*' variable, which are '*Biscoe*,' '*Dream*,' and '*Torgersen*,' as well as for the '*female*' and '*male*' categories of the '*sex*' variable. Once the encoding was completed, I concatenated the encoded variables to obtain my final penguins Dataframe.

From the Correlation Matrix in **Plot 7**, I noticed a significant positive correlation of 0.87 between the features '*body mass*' and '*flipper length'*. Additionally, there is a negative correlation of -0.75 between the features "Biscoe Island" and "Dream Island." This observation aligns with the findings from the previous plot, where I observed that these two islands each exclusively host a single penguin species (Gentoo species only on Biscoe Island and Chinstrap species only on Dream Island). Lastly, there is a perfect negative correlation between the features '*female*' and '*male'*.

As next step, I separated the features (X) from the label '*species*' (y) and split my dataset into training and test sets with a **70/30 ratio**. Next, I applied standardization to rescale my data. Standardization involves transforming all values in the dataset such that the mean value becomes 0 and the standard deviation becomes 1. This standardization process ensures that all the data points in my dataset have a consistent scale, making them easily comparable. With the data prepared, I then imported all the necessary modules from the *scikit-learn* library to create my machine learning models.

To develop my **K-Nearest Neighbor (KNN)** machine learning algorithm, I first plotted the accuracy of different k values. This helped me determine that a k value of 3 was the most suitable for my dataset, as the accuracy remained stable or didn't improve beyond this point. Next, I proceeded to train the KNN algorithm using 70% of the dataset and tested it on the remaining 30% of unseen data.

To ensure a robust evaluation, I performed both **5-Fold Cross Validation** and Stratified **10-Fold Cross Validation**. In 5-Fold Cross Validation, the data is randomly divided into 5 folds, without considering the class distribution, and each fold is used as a test set once while the rest act as the training set. On the other hand, Stratified 10-Fold Cross Validation maintains the class distribution in each fold, making it ideal for imbalanced datasets. Both methods train and evaluate the model multiple times, providing an average performance estimate. For my dataset, I achieved an average accuracy of 0.99 using 5-Fold Cross Validation and a slightly lower accuracy of 0.98 using Stratified 10-Fold Cross Validation.

To gain deeper insights into my machine learning model's performance, I created a dataframe containing the *y_test* column with the actual records, *y_pred* column with the predicted records, and other columns representing the probabilities of the predictions. This allowed me to analyze the model's correctness or incorrectness on a row-by-row basis.

Next, I generated a **confusion matrix** to evaluate my KNN model. All values in the matrix turned out to be true positives, indicating that the model perfectly predicted the data and classified them into the correct categories. Specifically, 48 instances were correctly classified as Adelie penguins, 18 instances as Chinstrap penguins, and 34 instances as Gentoo penguins.

To further validate the model's performance, I examined the classification report, which revealed perfect scores of 1 for **precision**, **recall**, and **F1-score**. These results provide additional confirmation of the model's accuracy. However, it is important to highlight that achieving such a level of perfection could raise concerns about potential overfitting. In reality, the model might

encounter different problems, and further analysis is necessary to comprehend the behavior of the machine learning algorithm effectively.

As the final step, I implemented the **Naïve Bayes** algorithm using a Gaussian distribution. However, it turned out to be a weaker model when compared to K-nearest neighbors (KNN). Both 5-Fold Cross Validation and Stratified 10-Fold Cross Validation revealed an average accuracy of only 0.72. Taking a closer look at the evaluation metrics, it can be observed that the Naïve Bayes algorithm achieved a perfect precision score of 1 for the Adelie species but had a very low recall score of 0.35, resulting in an overall F1 score of only 0.52. On the other hand, the F1 scores for the Chinstrap and Gentoo species were 0.72 and 0.80, respectively.

In conclusion, when working with this dataset, the K-nearest neighbors (KNN) algorithm outperforms the Naïve Bayes algorithm in accurately classifying the data. However, it is essential to exercise caution and thoroughly investigate potential issues like overfitting that might arise.

Dataset link: https://github.com/allisonhorst/palmerpenguins/blob/main/README.md