Work Documentation for Iris Dataset Analysis

Overview

There are three functions in the script, clean_data, find_relationship and build_model. The main program loads the iris dataset from Scikit learn package and save the data into pandas at the beginning. Then, it uses `df.describe()` to generate descriptive statistics of the dataset and store the info into a variable named `iris_description`. After that, it calls clean_data, find_relationship and build_model sequentially to generate all the graphs and data we want.

Cleaning Data

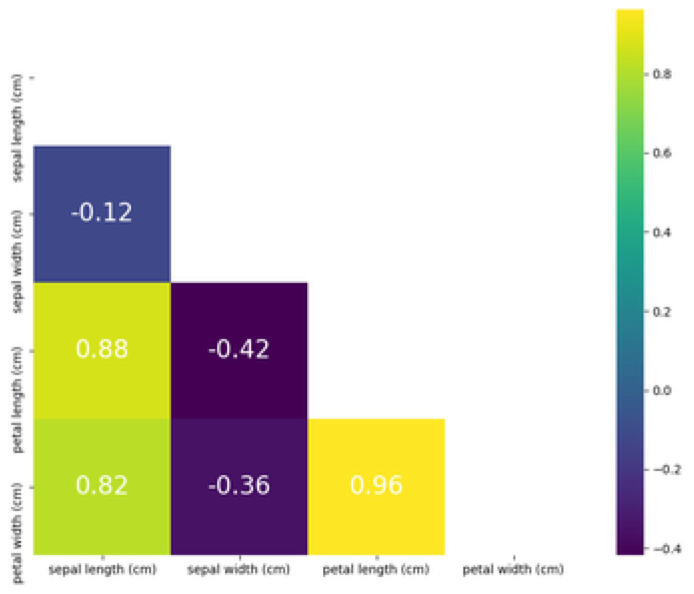
Function: `clean_data(df, z_thresh=3)`

This function takes data from pandas as input and removes all the rows with missing features. It also removes the data that are not within +3 to -3 standard deviations from the mean in the column `'sepal length (cm)'`, `'sepal width (cm)'`, `'petal length (cm)'` and `'petal width (cm)'`.

Find Statistical Relationships

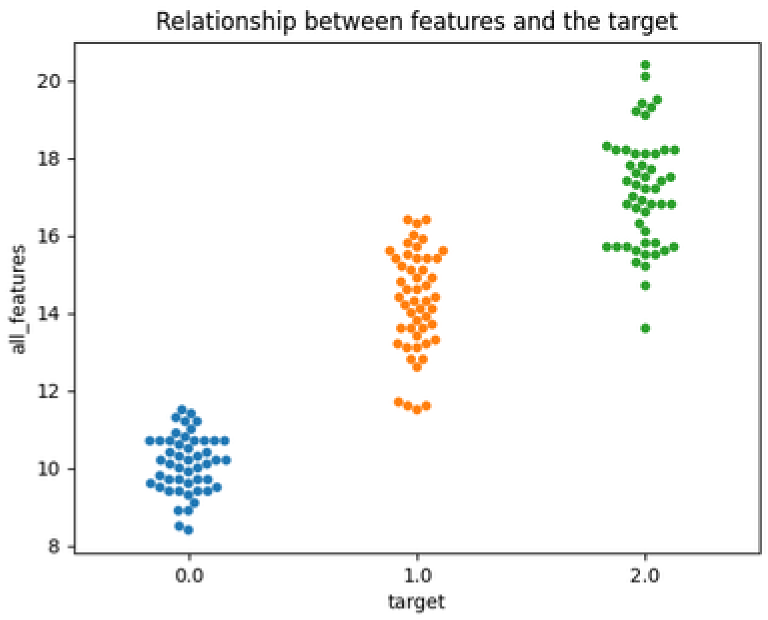
Function: `find_relationship(df):`

This function takes data from pandas as input and generates a correlation matrix in the format of a heat map to see if parameters correlate to each other.



As we can see from this heat map, the petal length and width are strong and positively correlates to each other. In contrast, there doesn't seem to be a relationship between sepal length and width. Additionally, sepal length also has a strong positive correlation with both petal width and length, and sepal width has a weak negative correlation with them.

This function also tries to find a relationship between features and targets. It adds up all the features to see if there is a relationship between the summation and the flower species. It saves the data into `Relationship.csv` and generates a swarm plot as the following.

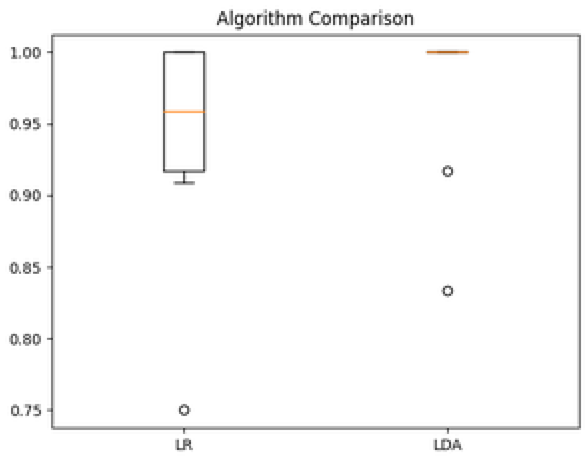


Target 0, 1, 2 are Iris Setosa, Iris Versicolor and Iris Virginica. This graph illustrates that Setosa is generally smaller in size and Virginica is bigger. Versicolor usually has a size between Setosa and Virginica.

Predict Targets with Different Models

Function: `build_model(df)`

This function takes data from pandas as input and utilizes it to create 2 models. First, it takes 20% out of the dataset for validation purposes. I didn't apply any normalizer because all the variables are in the same scale. Since the data has been cleaned, I believe that linear algorithms should be enough to build the model. So, I selected logistic regression and linear discriminant analysis and evaluate them. The evaluation result is saved in `Algorithm_Comparison.csv` and a box plot is generated based on 10 fold-cross validation.



We can see that the box and whisker plot for LDA is squashed at the top, with many evaluations achieving 100% accuracy, and some pushing down into ~85% accuracies. The LR model has good accuracy mostly from 90%~100% and one at ~75%, but not as good as LDA.