|  |  |
| --- | --- |
| Nguyen Manh Duy  Mail: manh.nguyen5@mail.dcu.ie  Student ID: 19214315 | Ninh Van Tu  Mail: van.ninh2@mail.dcu.ie  Student ID: 18214653 |

Iris Problem

In this project, we want to create the prediction model that can classify Irish flower species (**setosa**, **verginica**, **versicolor**) based on 4 features including **petal length**, **petal width**, **sepal length**, **sepal width**. Our workflow is as followed:

* Checking the data sanity, e.g Are there any missing values in the dataset?
* Cleaning dataset, e.g make the prediction of any missing value
* Analyzing feature
* Building the model

# 1. Checking Sanity

There are 50 samples for each species in the dataset. Setosa class has 1 sample not having the feature sepal length, while versicolor species has 1 missing value in each sepal width and petal length feature. The verginica category is clean.

# 2. Cleaning Data

We perform linear regression to fill in the missing value. For each regression model, we only use the data from the species containing missing value. We also split the data into training (80% of data) and testing (20%) subsets to evaluate the model.

## Setosa

The model will predict Sepal.Length based on other features. Before doing the regression, we need to check the correlation between features in order to decide which features to be used in the regression model (Fig 1).

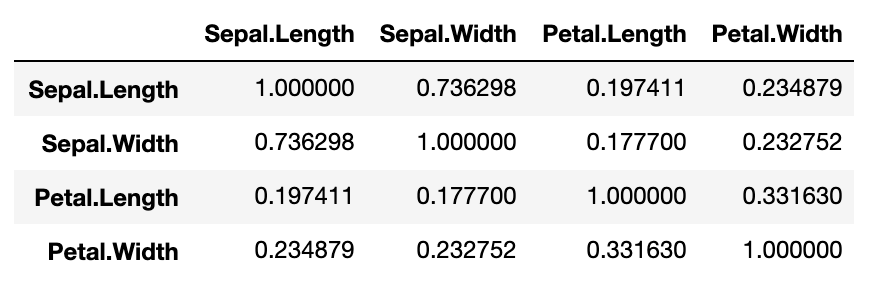


Figure 1. The correlation coefficients between features   
(you may hate it but most of our analysis is based on this)

The figure indicates that we may need only Sepal.Width feature to create the model. However, we still use all 3 features to make the comparison whether our assumption is correct. In Python we easily do it with *statsmodels.formula.api* library and it also gives us insight information about the created model.

For the model using all 3 features, we achieve and the p values for intercept, Sepal.Width, Petal.Length, Petal.Width are given as 0, 0, 0.77, and 0.44 respectively. It tells us that we can discard 2 last features since their coefficient are not significantly different from zero.

The result from the model using only Sepal.Width feature is also consistent with the above conclusion. Although it has lower , lower values of prove that this simple model is better than the model using all features. Besides, the residual scatter plot (Fig 2) shows that there is no sign of overfitting since the error on the testing set is similar to that on training.

## Versicolor

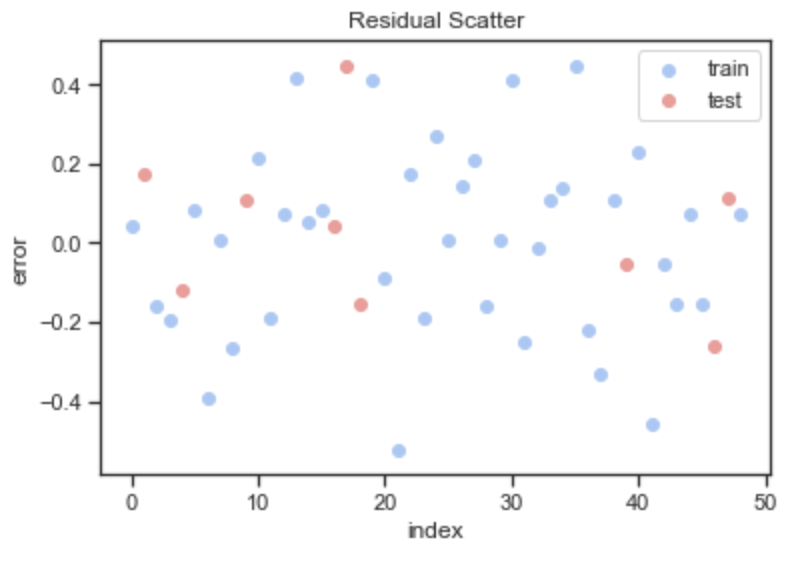


Figure 2. Residual Scatter plot

We build 2 distinct regression models to predict the missing values in 2 feature sepal width and petal length. With the same steps as above section, we finally have the model of and . More information can be found in our source code.

# 3. Analyzing Feature

The interpolated data now is splitted into 2 subset: training and testing to ensure the generality of our classification. The analyzing step is only performed within the training set and we leave the testing set out. Here we only plot the boxplots of 4 features of 3 species to have some knowledge to build the final prediction model.

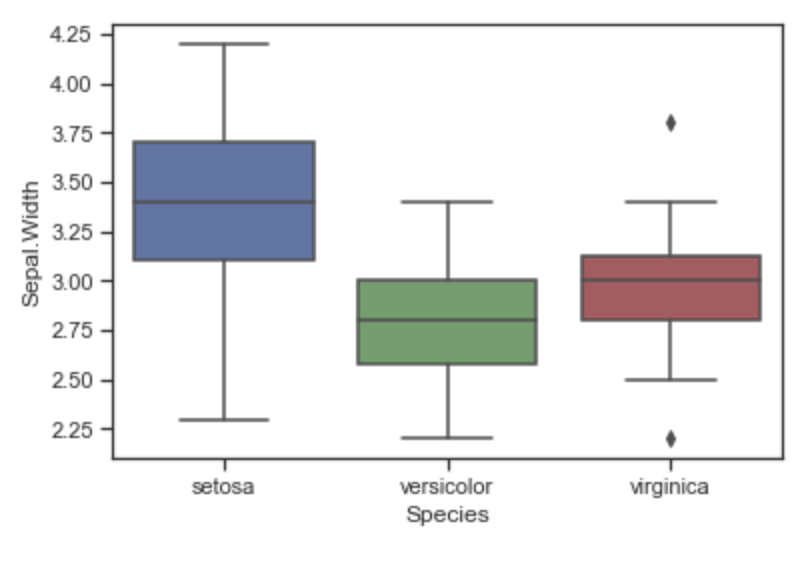
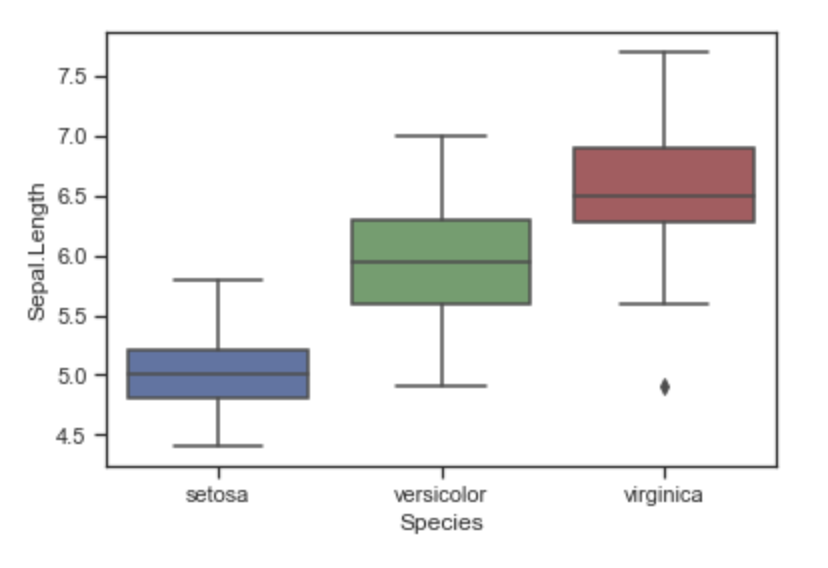
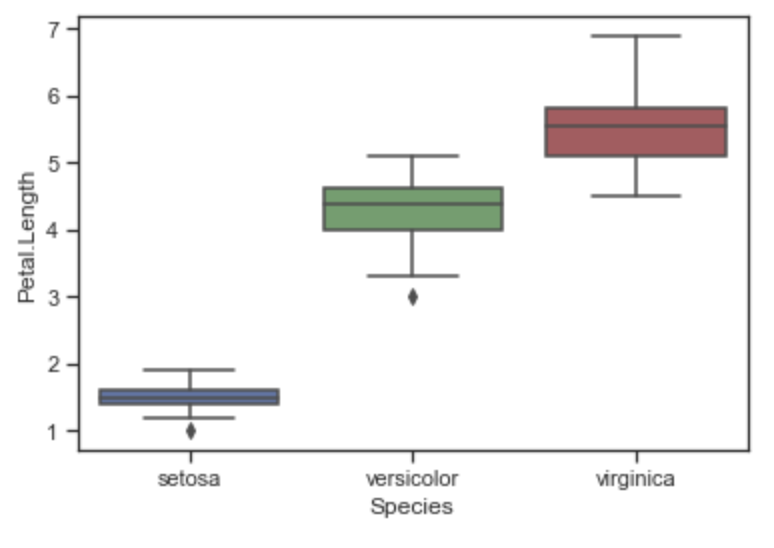
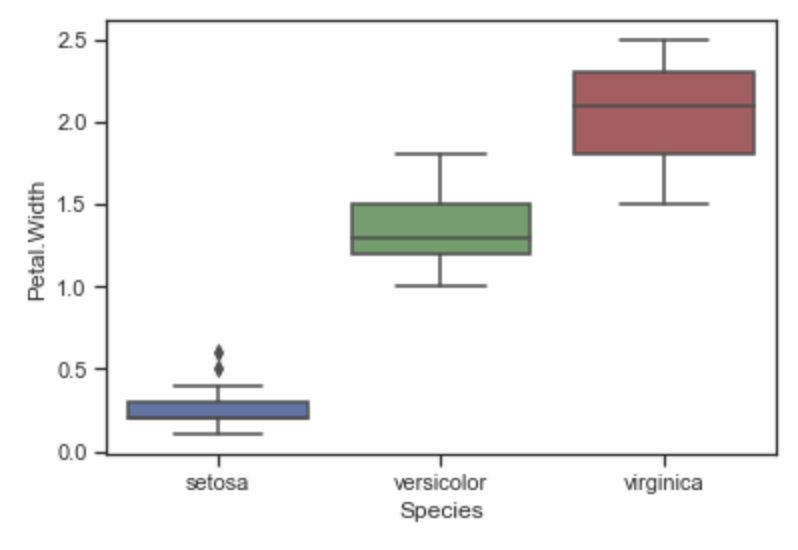


Figure 3. Distribution of features within each class

Fig 3 tells us that petal length (or petal width) features of setosa class are totally smaller than those from the other 2. With the simple rule, such as , we can easily detect setosa species and change the 3-category classification into binary classification.

With the simple approach, we will try logistic regression to differentiate versicolor and virginica species. Fig 3 also shows that the petal length and petal width feature may have the most contribution to our model (it is our assumption).

# 4. Building prediction model

We use *sklearn.linear\_model* library to build the logistic model. 80% samples of each species are used as training set and 20% for testing. The result coefficients from logistic regression indicate that our assumption is correct. The accuracy of the final model, which combines rule-based and logistic regression, on training, testing, and entire dataset are 0.975, 0.96, and 0.973 accordingly, meaning that the final model does not meet overfitting problem. The confusion matrix and other figures can be found in our source code, which might be updated (<https://github.com/m2man/CA683>)