Analysis of Logistic Regression on IRIS Dataset

PREPARED FOR

ML Assignment 02

PREPARED BY

Group 31

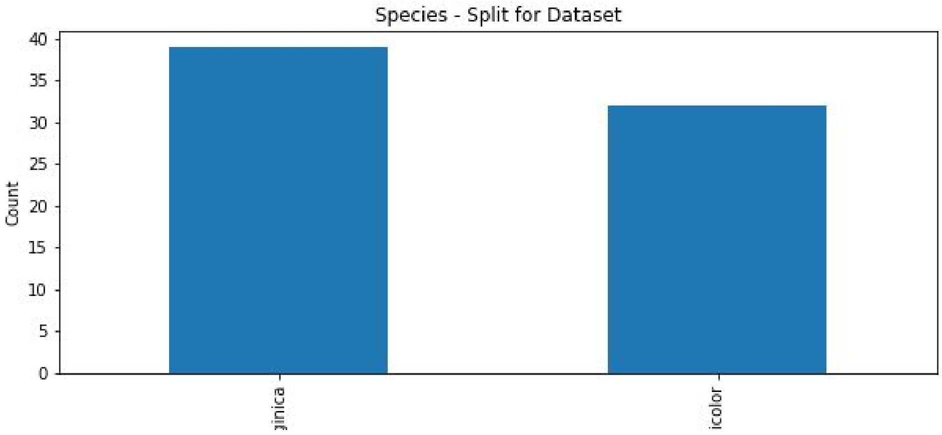
Aditi Verma (S20180010006)
Nethra Gunti (S20180010061)
Kavya Nemmoju (S20180010078)
Dasari Jayasree (S20180010047)

Abstract

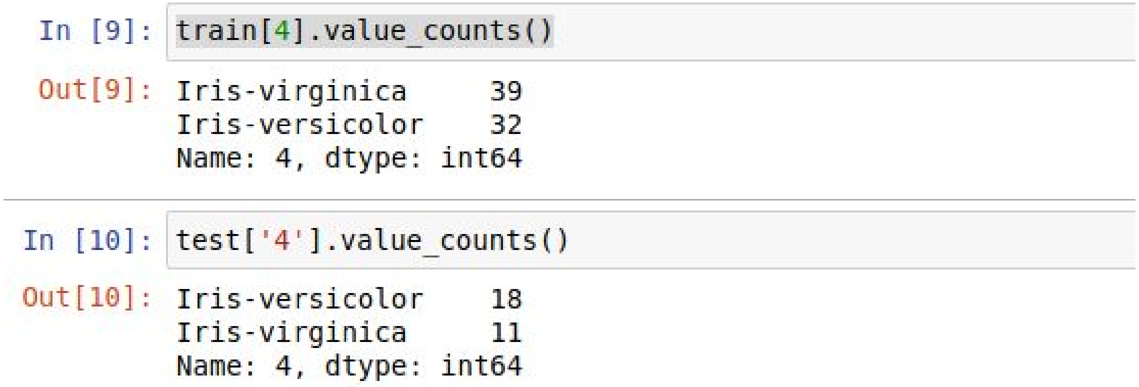
The aim of this project is to implement Logistic Regression for the given IRIS dataset. Given the 4 featured samples, we have to predict the classes from ‘Iris Virginica’ and ‘Iris Versicolor’. The dataset contains 99 samples with 70-30 train-test ratio. In this statistical model, we have used logistic functions to model the binary dependent value inorder to classify the given.

Exploratory Data Analysis (EDA)

Before making the Logistic Regression model, we decided to perform EDA on the training dataset to gain more insight into what we’re working with. In this section, we cover the results.

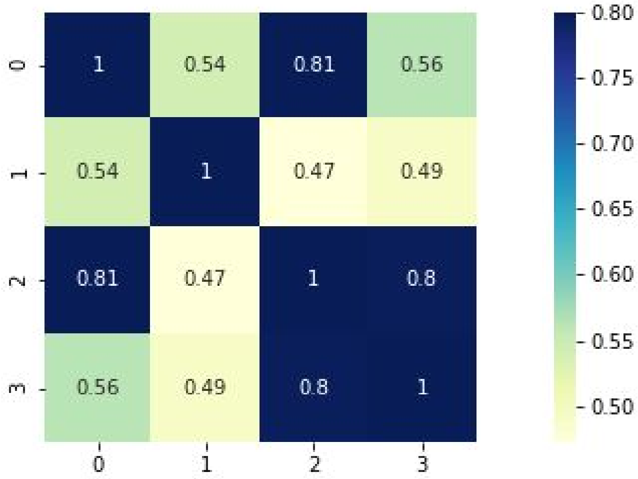


Here, we can see the number of samples of each species present in the training dataset.

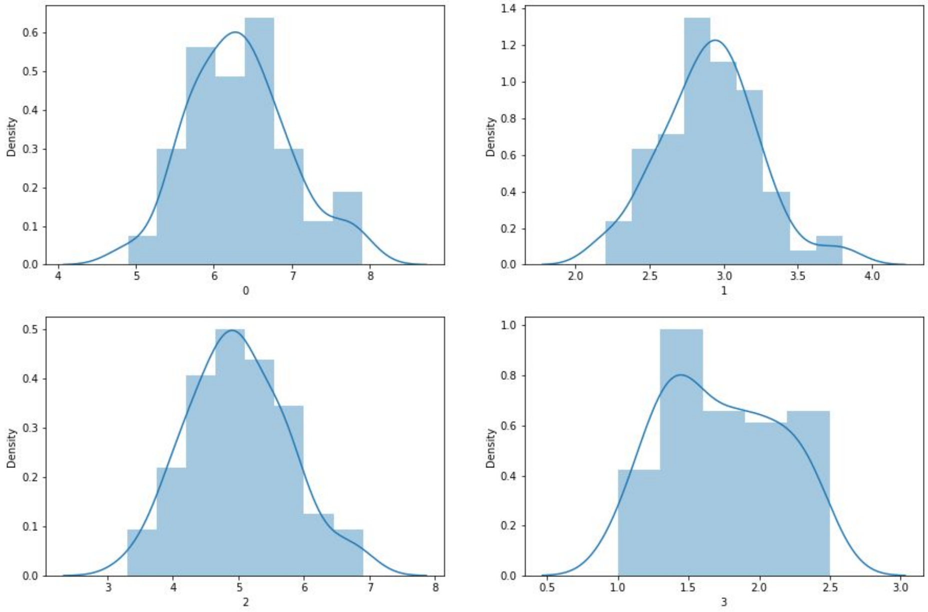


```
In [9]: train[4].value_counts()

Out[9]: Iris-virginica     39
        Iris-versicolor    32
        Name: 4, dtype: int64
```

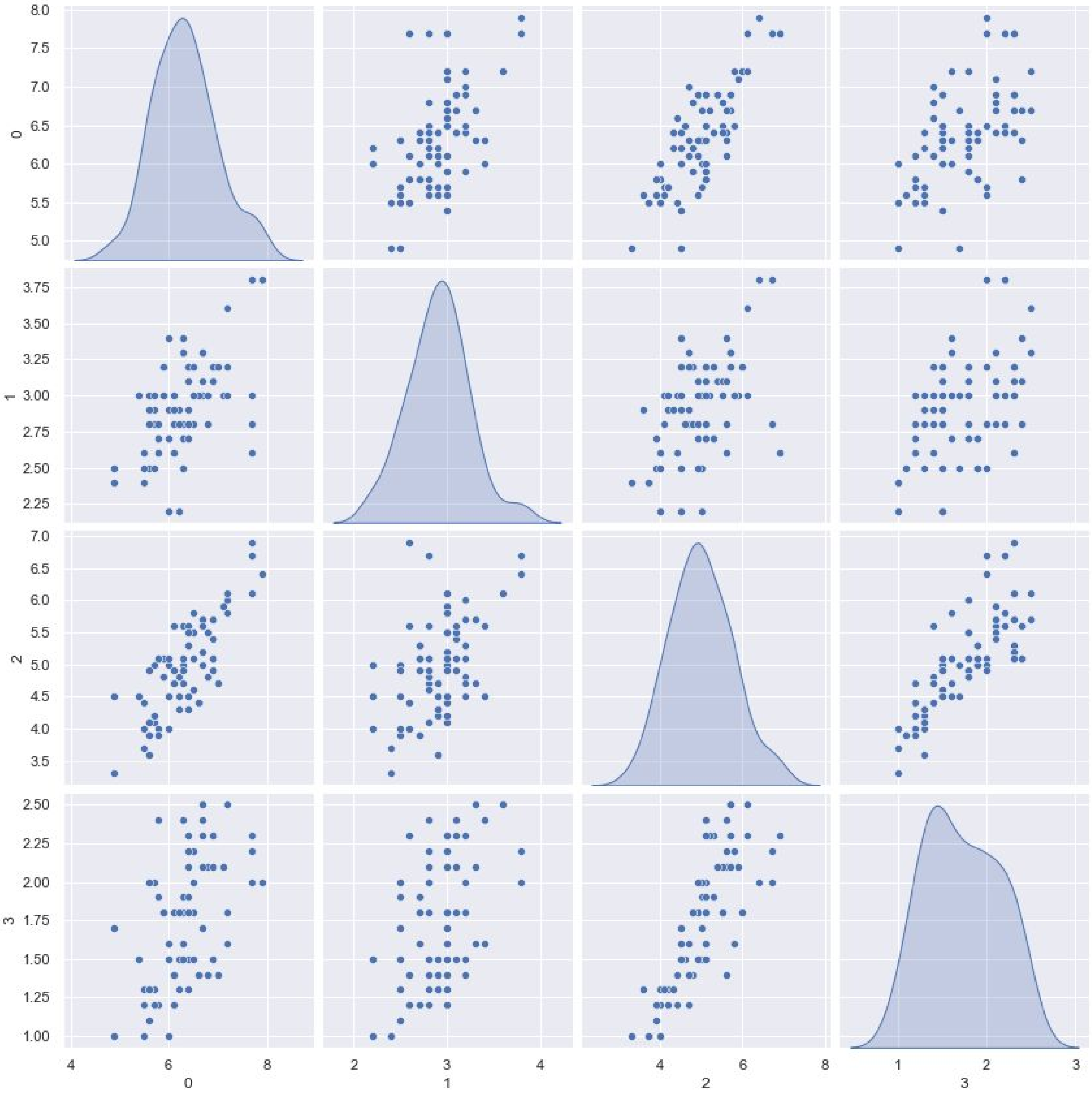
```
In [10]: test['4'].value_counts()

Out[10]: Iris-versicolor    18
         Iris-virginica     11
         Name: 4, dtype: int64
```



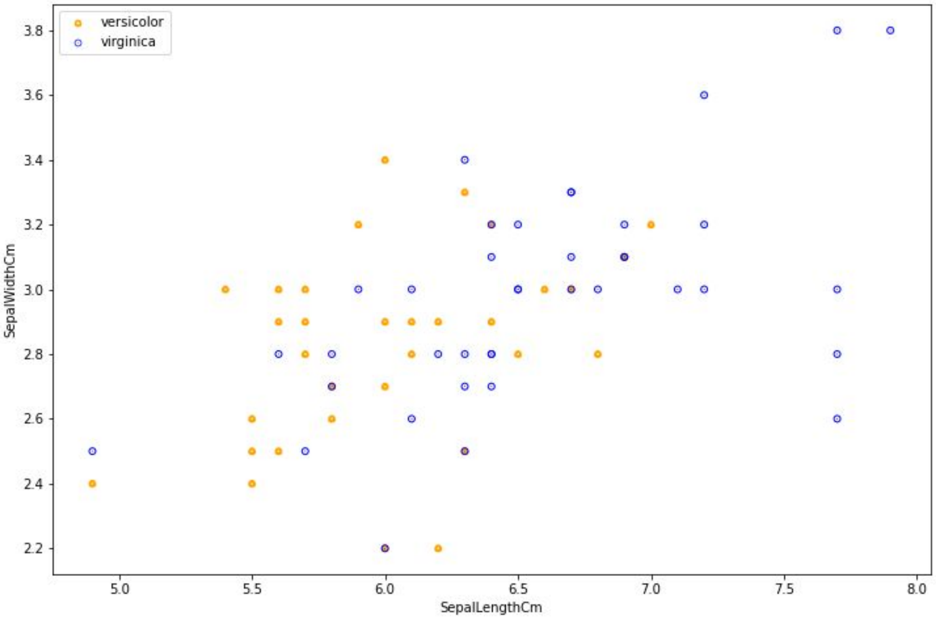
Here, we see the correlation heatmap between the different features of the dataset. We notice that columns 0 and 2 and 2 and 3 have particularly high correlations.

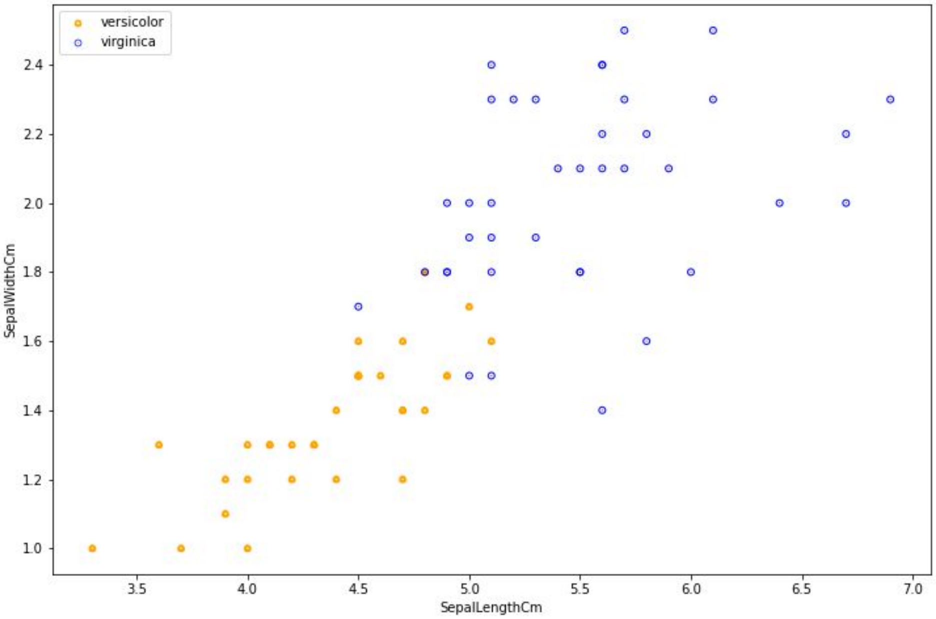


This plot shows the distribution of the various features.

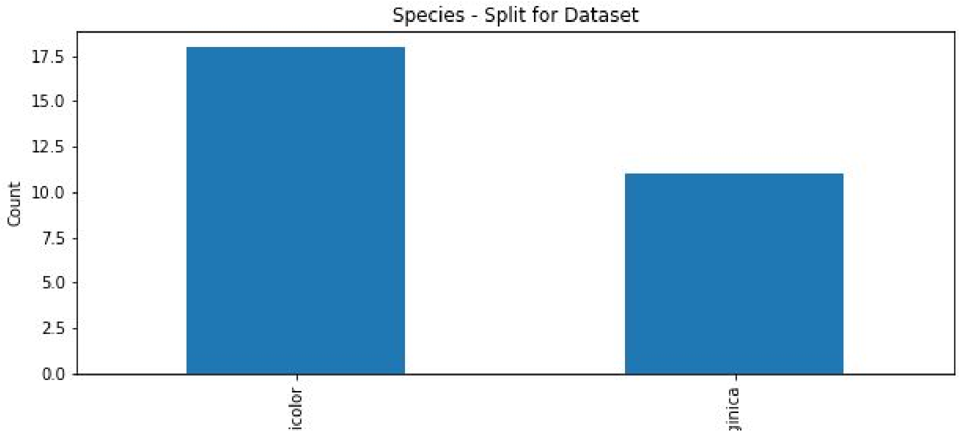


This is the pair plot for the features in the training set.

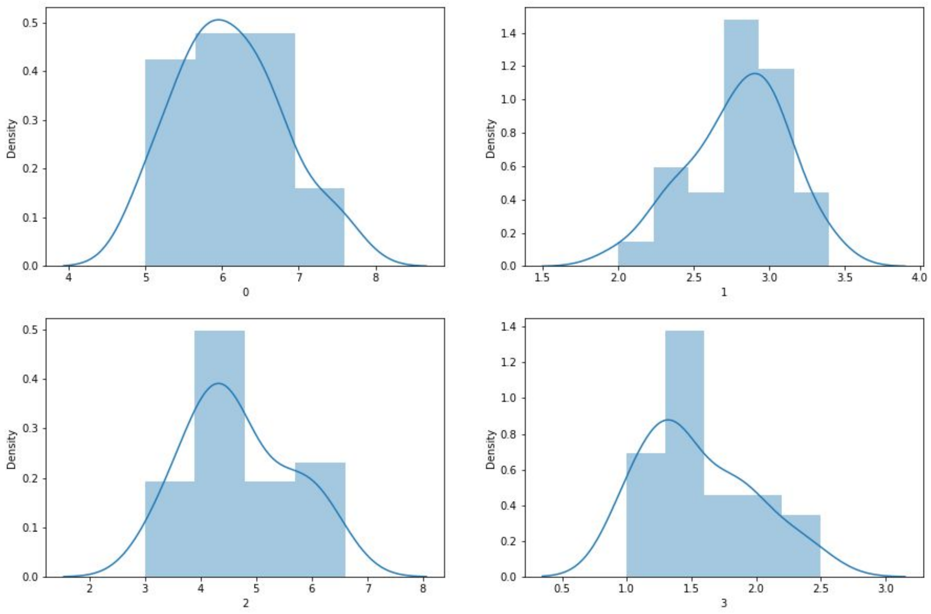




The bivariate distribution for features in the train dataset.



This shows the count of the test dataset.



The distribution of the features of the test dataset.

Approach

From the given split train and test datasets, we first train the model with the training set and predict for the test dataset. We normalise the feature vectors in both the datasets (x_train, x_test).

$$f_{normalised} = \frac{f_{original} - f_{mean}}{f_{standard\ deviation}}$$

Given the two class problem, we assign the classes a value of 0 and 1. We calculate the dot product using the augmented feature vector X and the weight vector θ which is initialized with zeros of size n+1, where n is the number of features.

$$h(x) = s(\Sigma_{i=0}^{d} \theta_i x_i)$$

i.e, $h(X) = s(X \times \theta)$

Inorder to classify the sample, we pass the product to the sigmoid function s(x) which gives a scalar values between 0 and 1.

$$s(x) = \frac{1}{1+e^{-x}}$$

The sample with this predicted value if less than 0.5, is classified as class 0, otherwise class 1. The loss c(x) of this prediction is given by using the *Binary Cross Entropy*.

$c(x) = - y \, log(h(x)) - (1 - y) \, log(1 - h(x))$,where y is the original class in the dataset and h(x) is the predicted value.

For utmost accuracy, we update our weights until we have reached minimum error or have fulfilled the specified stopping criterion.

$$\theta_{new} = \theta_{old} + \eta \Delta J \ ,$$

where, $\Delta J = X^{T} \times (Y - h(X))$

where ΔJ is the gradient of the loss with respect to the weights and η is the learning rate of the model.

$$\theta_{new} = \theta_{old} + \eta \, \Sigma_{i=1}^{n} (y^{i} - h^{i}) x^{i}$$

The first case, instead of fixing the number of epochs, a stopping criterion of $||\Delta J|| < 0.01$ has been used , where $||\Delta J||$ is the Euclidean norm of vector ΔJ .

In the second case, the number of epochs have been fixed to 80.

Results

The results that we got from this modeling are as follows:

Case 1: $\triangle J < 0.01$

Number of epochs: 2964

Loss: 5.474326

Training set accuracy: 97.183%

Test set accuracy: 100%

Case 2: Fixed number of Epochs

Number of epochs: 80

Loss: 7.231752

Training set accuracy: 97.183%

Test set accuracy: 100%

Resources:

Slides from the class provided by Viswanath sir

https://en.wikipedia.org/wiki/Sigmoid_function

https://en.wikipedia.org/wiki/Logistic_regression

https://machinelearningmastery.com/