

NAME: SAMRIDDHI VERMA
REG. NO.: 16BCE1375
SLOT: G1
MACHINE LEARNING

DIGITAL ASSIGNMENT-1

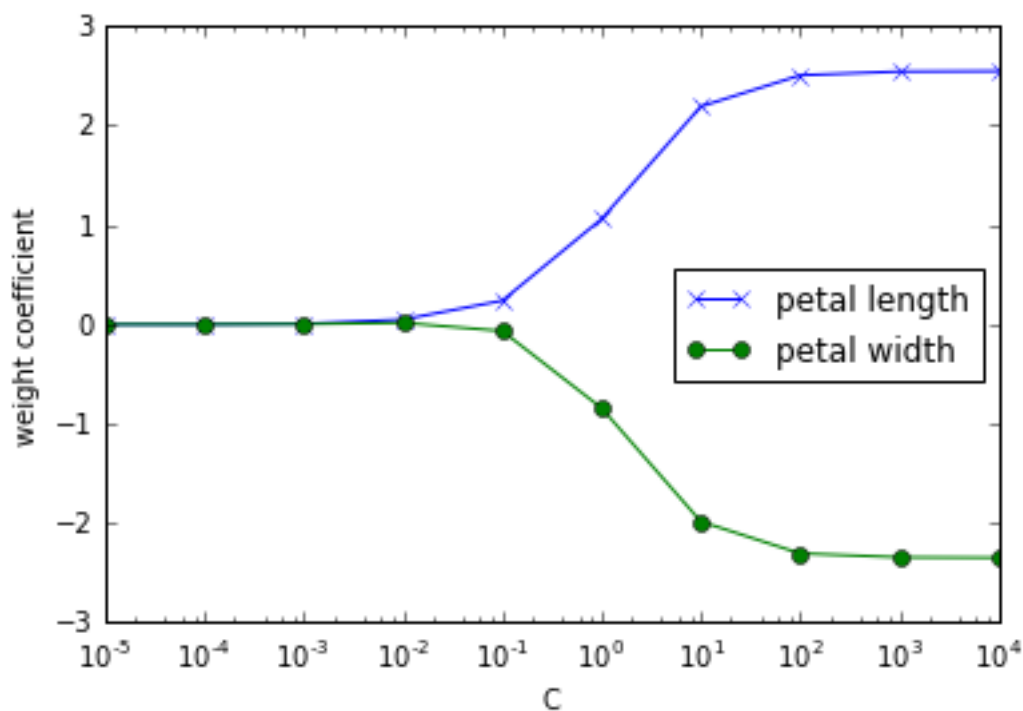
PROBLEM:

IMPLEMENT AND COMPARE THE RESULTS OF ANY ML ALGORITHM AND REGULARIZATION AND WRITE INFERENCES.

SOLUTION:

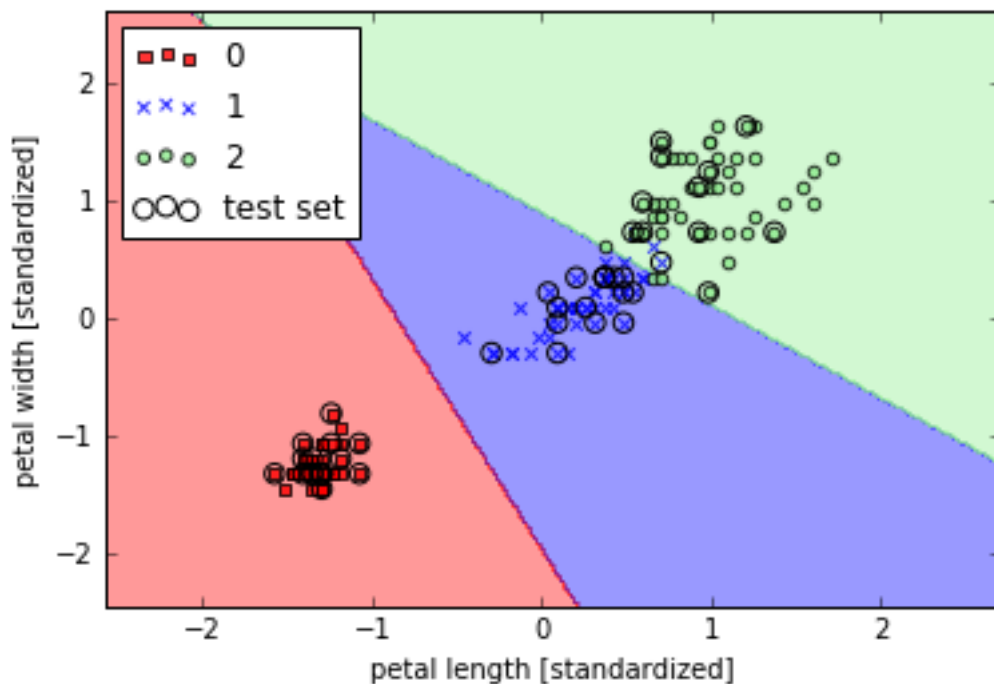
I HAVE IMPLEMENTED LOGISTIC REGRESSION ON THE IRIS DATASET AND ALSO REGULARISATION ON THE SAME DATASET. THE CODES FOR THEM ARE IN THE RESPECTIVE PYTHON FILES AND OUTPUTS ARE BELOW:

REGULARIZATION:



NAME: SAMRIDDHI VERMA
REG. NO.: 16BCE1375
SLOT: G1
MACHINE LEARNING

LOGISTIC REGRESSION:



INFERENCES ABOUT THE RESULTS:

For the iris-dataset, as we've done before, we split the set into separate training and test datasets: we randomly split the X and y arrays into 30 percent test data (45 samples, index 105-149) and 70 percent training data (105, index 0-104) samples. We also did feature scaling for optimal performance of our algorithm using the StandardScaler class from scikit-learn's pre-processing module. Also, by using the fit method, StandardScaler estimated the parameter of sample mean and standard deviation for each feature dimension from the training data. Then, by calling the transform method, we standardized the training data using those sample mean and standard deviation. For the testing data, we used the same scaling parameters to standardize the set so that both the values in the training and test dataset are comparable to each other. In addition to that our model can suffer from under-fitting or even over-fitting. To avoid this, we use additional techniques e.g. cross-validation, regularization, early stopping, pruning, or Bayesian priors. To apply regularization to our logistic regression, we just need to add the regularization term to the cost function to shrink the weights. Via the regularization parameter λ , we can then control how well we fit the training data while keeping the weights small. By increasing the value of λ , we increase the regularization strength.

NAME: SAMRIDDHI VERMA
REG. NO.: 16BCE1375
SLOT: G1
MACHINE LEARNING

The parameter C that is implemented for the LogisticRegression class in scikit-learn comes from a convention in support vector machines, and C is directly related to the regularization parameter λ , which is its inverse, i.e., $C = 1/\lambda$. As we can see in the following plot, the weight coefficients shrink if we decrease the parameter C (increase the regularization strength, λ).