A Minor Project

on

CLASSIFICATION ANALYSIS ON IRIS DATA

The classification of iris dataset in this project is performed using Support Vector Machine (SVM) Algorithm using linear kernel as well as non-linear kernel i.e., Radial Basis Function (RBF) kernel and its resulting accuracy is compared.

Tools used :

* Python – The code of the program is written in python programming language using packages and libraries such as Pandas, NumPy and Matplotlib etc
* Google Colaboratory -- Colaboratory, or “Colab” for short, is a product from Google Research where we can write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. In this project I have used this platform for coding.

Algorithm:

**Support Vector Machine (SVM) Algorithm**

SVM or Support Vector Machine is a linear model for classification and regression problems. The idea of SVM is simple: The algorithm **creates a line or a hyperplane which separates the data into classes**.

There are specific types of SVMs you can use for particular machine learning problems, like support vector regression (SVR) which is an extension of support vector classification (SVC). In this project we have used Support Vector Classification (SVC) using linear and radial basis function kernels.

**Linear kernel** is used when the data is Linearly separable, that is, it can be separated using a single Line. It is one of the most common kernels to be used.

Here's the function that defines the linear kernel:

f(X) = w^T \* X + b

In this equation, **w**is the weight vector that you want to minimize, **X**is the data that you're trying to classify, and **b**is the linear coefficient estimated from the training data. This equation defines the decision boundary that the SVM returns.

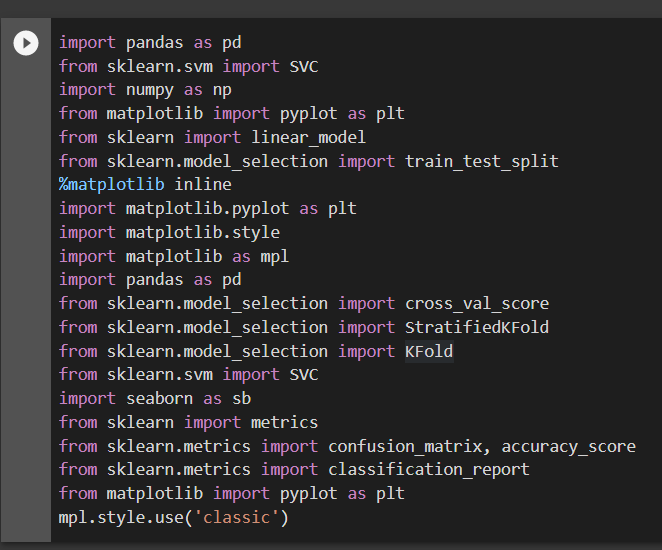
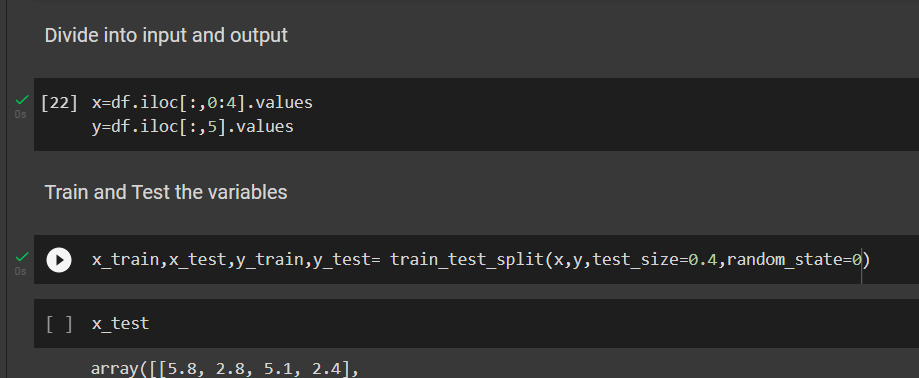
**Gaussian Radial Basis Function (RBF)** One of the most powerful and commonly used kernels in SVMs. Usually the choice for non-linear data.

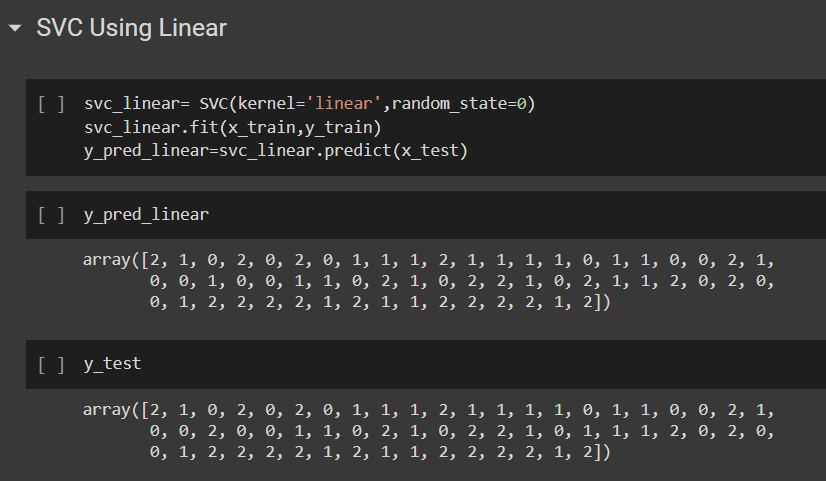
Here's the equation for an RBF kernel:

F (X1, X2) = exp (-gamma \* ||X1 - X2||^2)

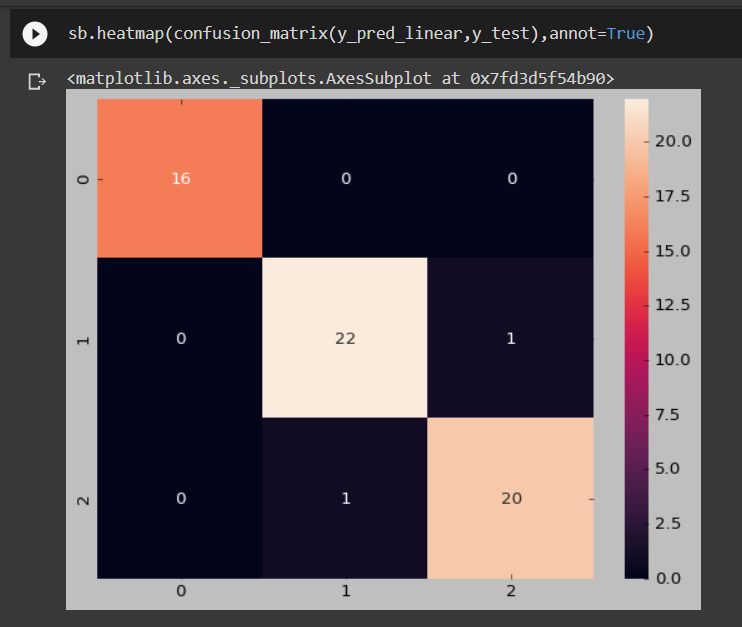
In this equation, gamma specifies how much a single training point has on the other data points around it. ||X1 - X2|| is the dot product between your features.

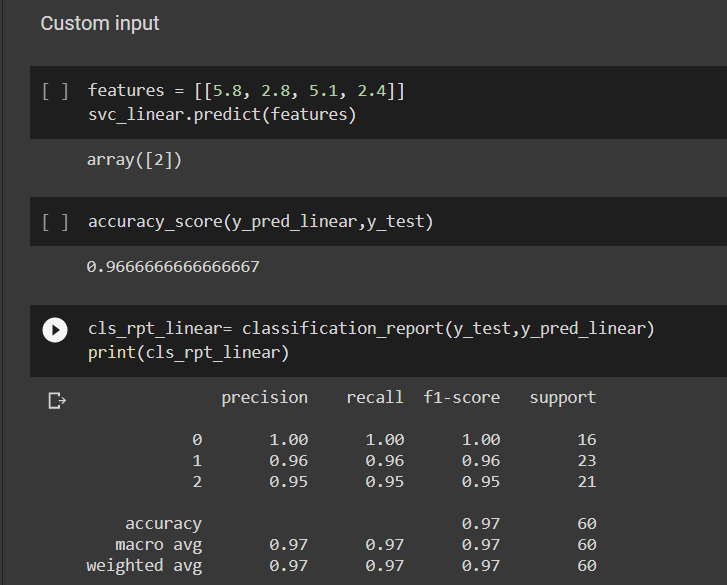
**Code Explanation**

* + Firstly we import all the neccessary packages and libraries
  + Then we upload the data file using “ pd.read\_csv(“File name or path”)”
  + We then Visualize the given data until the correlation is found
  + Now columns are divided into input and output to train and test them as follows by importing model selection from scikit.
  + Now by using svc classifier we need to train and fit the train variables to the model then we get the remaining predicted values.

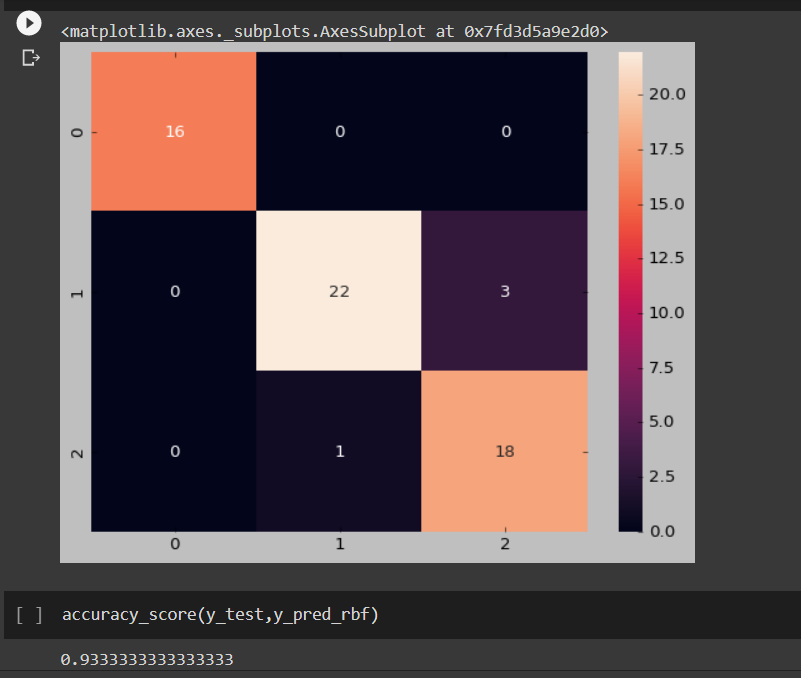


* + Now by using the predicted values we evaluate the model using confusion matrix,here I have represented it in pictorial way using heatmap() after this a custom input is given to check the output

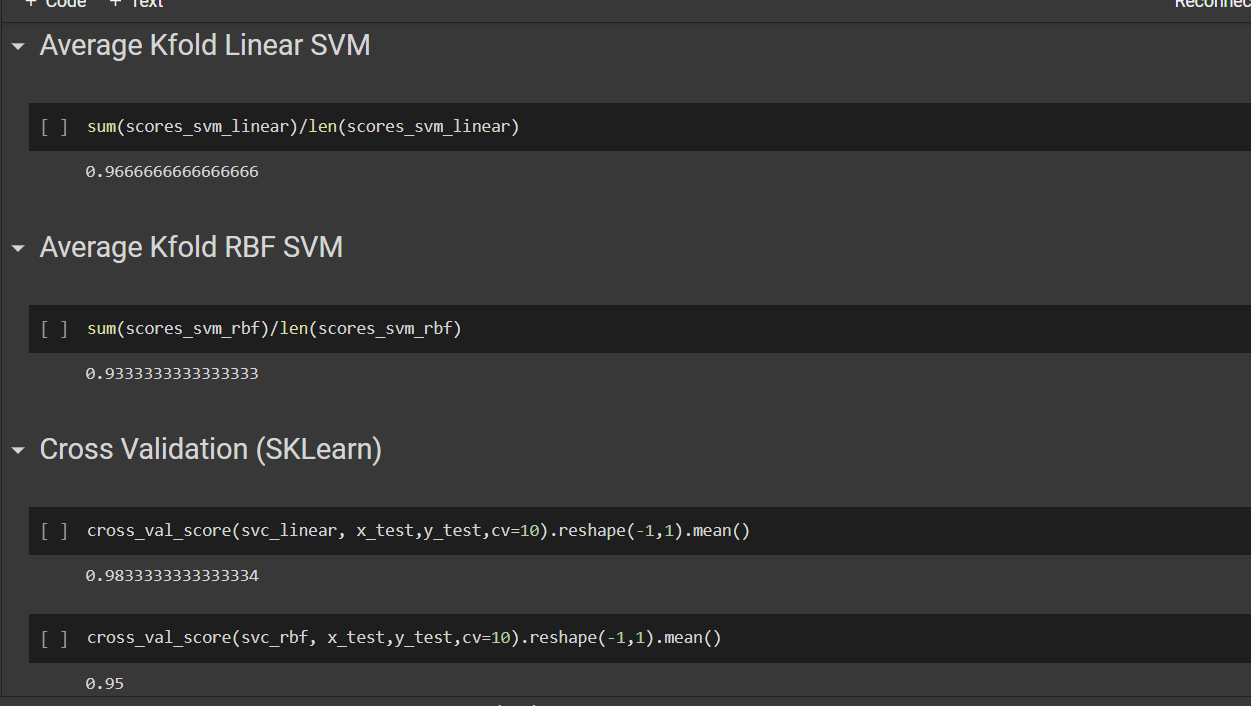




* + In the same way as we need to classify using the rbf kernel then we get the following results



* + By comparing the accuracy rate from linear kernel and rbf kernel it is observed that svc using linear kernel has produced better accuracy rate than rbf.
  + We can then cross validate the data using K- Fold Cross Validation



THE END