CSE 4710
Machine Learning Lab

Experiment No: 7

Name of the experiment: Implementing Support Vector Machine Classifier and Tuning Hyper-parameters

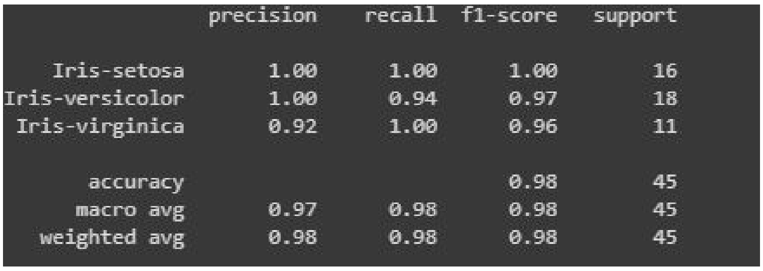
Hasan Mahmud
Assistant Professor, Department of CSE
Fardin Saad
Lecturer, Department of CSE

August 10, 2020

1 Tasks

- **Task 01:**
  Load the Iris Dataset given to you and split it with a 7:3 ratio for training and testing your SVM model.

- **Task 02:**
  Train the dataset with Linear Kernel SVM classifier and display its accuracy, confusion matrix and classification report (Precision, recall etc). Visualize the confusion matrix with a heat-map and correctly label it.
  Hint: Use sklearn library for classification report. The classification report and confusion matrix is given below:



```
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        16
Iris-versicolor       1.00      0.94      0.97        18
 Iris-virginica       0.92      1.00      0.96        11

       accuracy                           0.98        45
      macro avg       0.97      0.98      0.98        45
   weighted avg       0.98      0.98      0.98        45
```

Figure 1: Classifiaction report

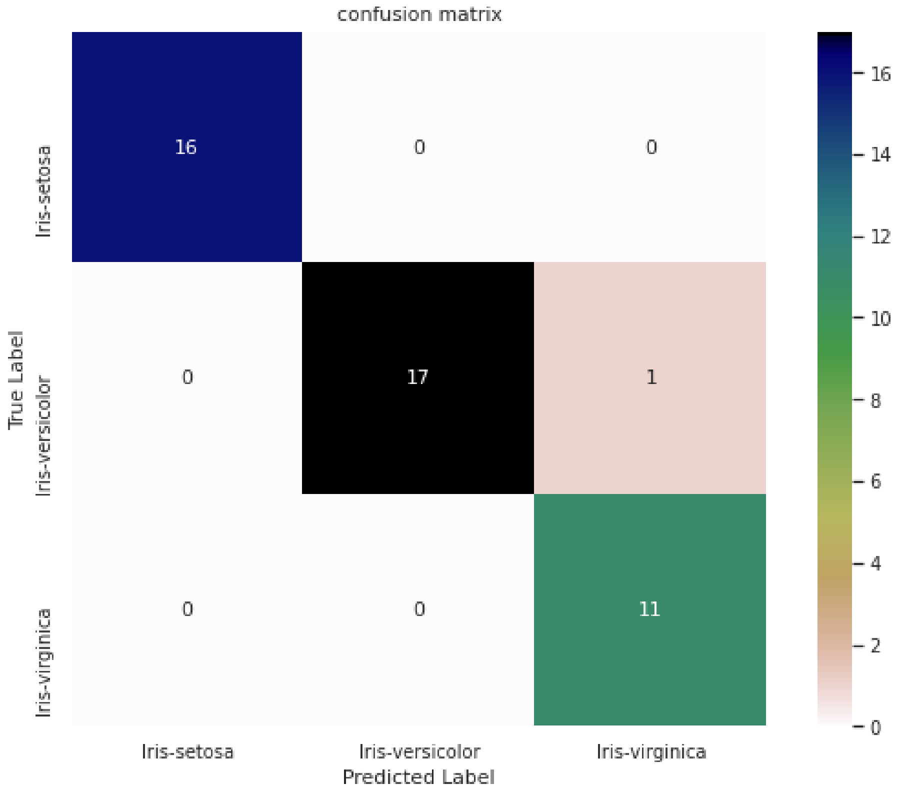


Figure 2: Confusion Matrix

- **Task 03:**
  Train the dataset with Polynomial Kernel SVM classifier and display its accuracy, confusion matrix and classification report. Visualize the confusion matrix with a heat-map and correctly label it.

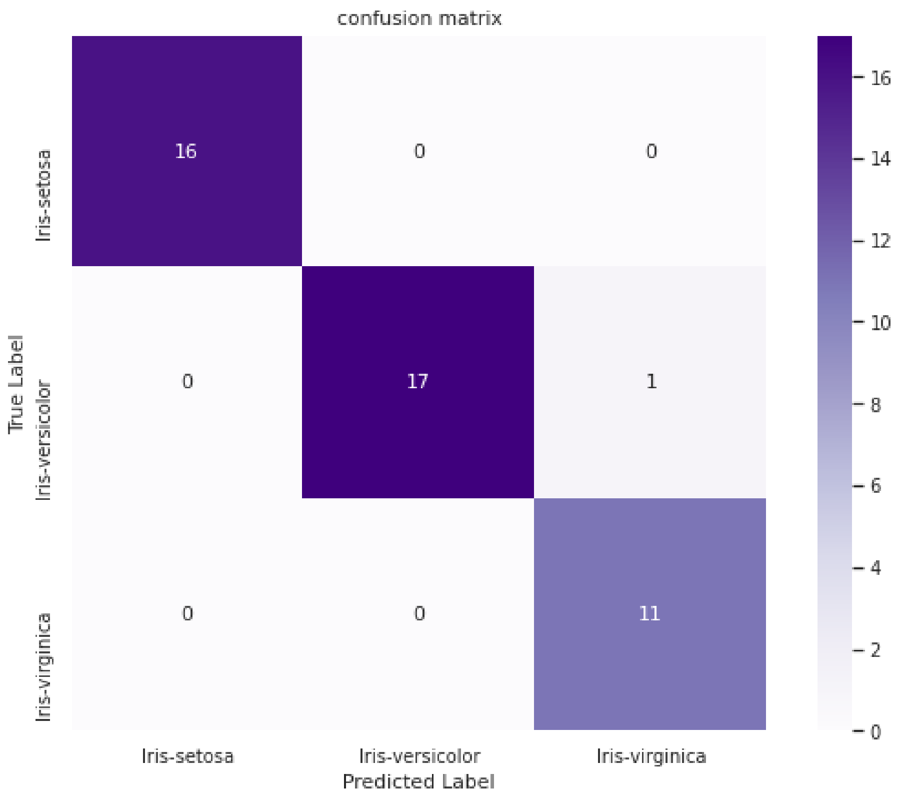
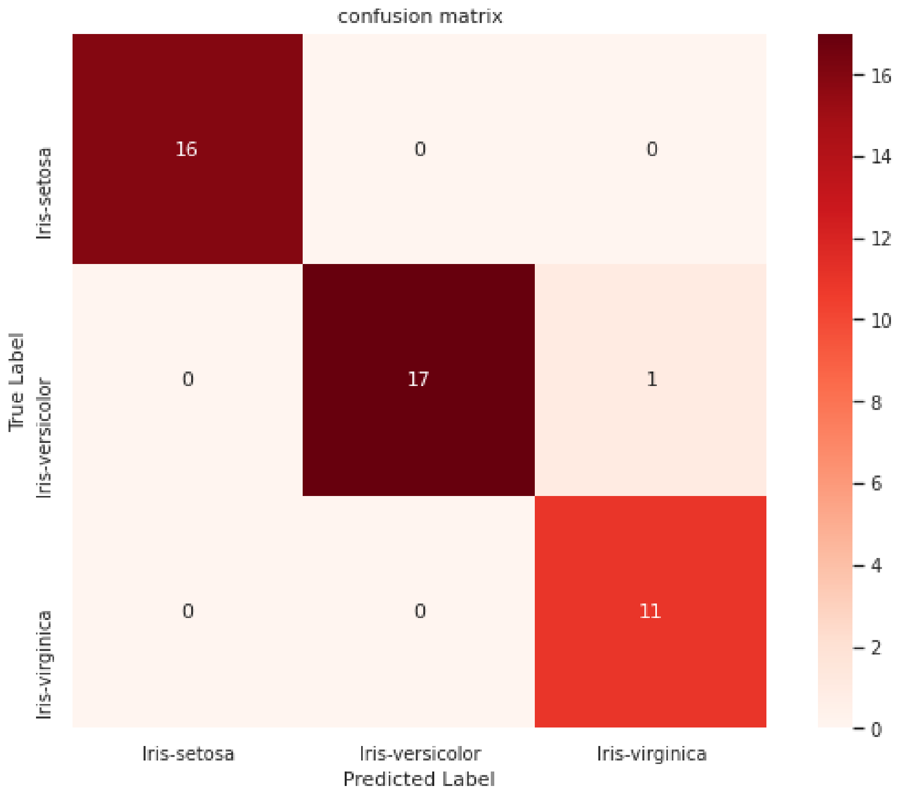
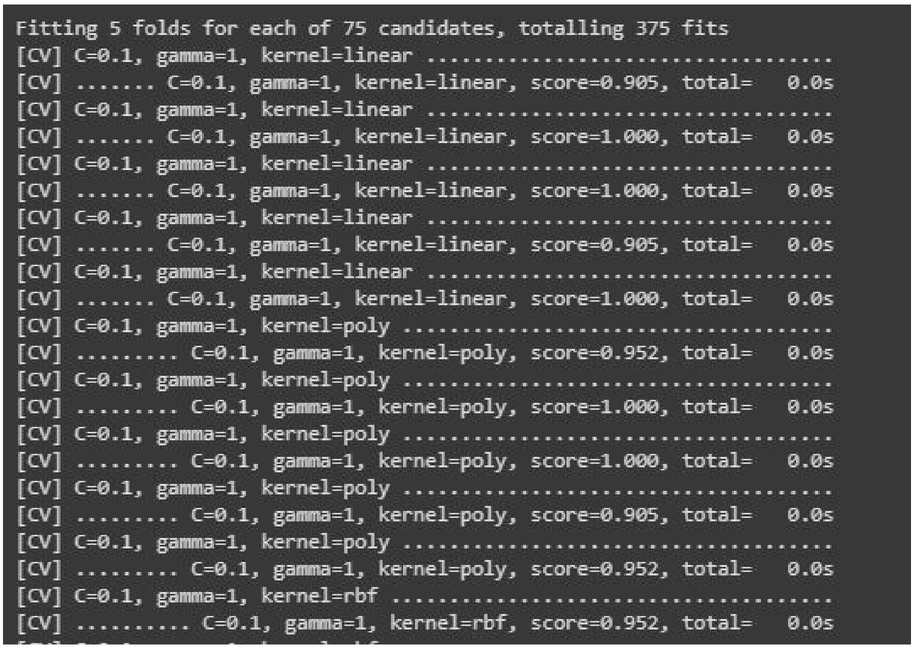


Figure 3: Confusion Matrix

- **Task 04:**
  Train the dataset with Gaussian Kernel SVM classifier and display its accuracy, confusion matrix and classification report. Visualize the confusion matrix with a heat-map and correctly label it.
  Hint: For Gaussian SVM use 'rbf' as kernel which is also known as Radial Basis Function.

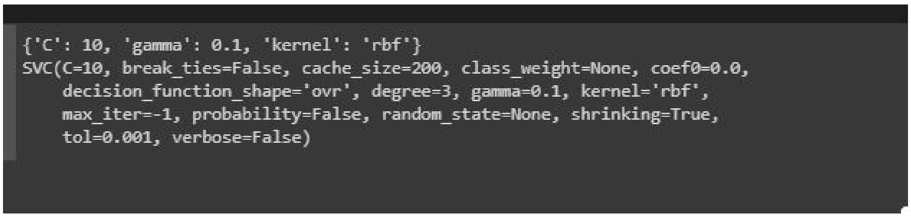


- **Task 05:**
  Tune your Hyper-parameters, C and Gamma using the Grid Search Cross Validation for all the kernels listed above and print the best parameters and estimators.



```
Fitting 5 folds for each of 75 candidates, totalling 375 fits
[CV] C=0.1, gamma=1, kernel=linear ...................................
[CV] ....... C=0.1, gamma=1, kernel=linear, score=0.905, total=   0.0s
[CV] C=0.1, gamma=1, kernel=linear ...................................
[CV] ....... C=0.1, gamma=1, kernel=linear, score=1.000, total=   0.0s
[CV] C=0.1, gamma=1, kernel=linear ...................................
[CV] ....... C=0.1, gamma=1, kernel=linear, score=1.000, total=   0.0s
[CV] C=0.1, gamma=1, kernel=linear ...................................
[CV] ....... C=0.1, gamma=1, kernel=linear, score=0.905, total=   0.0s
[CV] C=0.1, gamma=1, kernel=linear ...................................
[CV] ....... C=0.1, gamma=1, kernel=linear, score=1.000, total=   0.0s
[CV] C=0.1, gamma=1, kernel=poly .....................................
[CV] ......... C=0.1, gamma=1, kernel=poly, score=0.952, total=   0.0s
[CV] C=0.1, gamma=1, kernel=poly .....................................
[CV] ......... C=0.1, gamma=1, kernel=poly, score=1.000, total=   0.0s
[CV] C=0.1, gamma=1, kernel=poly .....................................
[CV] ......... C=0.1, gamma=1, kernel=poly, score=1.000, total=   0.0s
[CV] C=0.1, gamma=1, kernel=poly .....................................
[CV] ......... C=0.1, gamma=1, kernel=poly, score=0.905, total=   0.0s
[CV] C=0.1, gamma=1, kernel=poly .....................................
[CV] ......... C=0.1, gamma=1, kernel=poly, score=0.952, total=   0.0s
[CV] C=0.1, gamma=1, kernel=rbf ......................................
[CV] .......... C=0.1, gamma=1, kernel=rbf, score=0.952, total=   0.0s
```

Figure 4: Hyper-parameter Tuning



```
{'C': 10, 'gamma': 0.1, 'kernel': 'rbf'}
SVC(C=10, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma=0.1, kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

Figure 5: Best Hyper-paramters and Estimators