CSE303 (Section 1)

[Spring 2022]

Lab Assignment Submission Report

Assignment Title: LAB 08

Submitted by:

Rizvee Hassan Prito

2019-3-60-041

1.Screenshots

Model based on Support Vector Machine:

```
import pandas as pd
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn import metrics
import matplotlib.pyplot as plt

iris = datasets.load_iris()

X_train, X_test, Y_train, Y_test = train_test_split(iris.data, iris.target,
                                    test_size = 0.2, random_state = 42)

model = svm.SVC(kernel = 'linear', C = 10, probability = True)

model.fit(X_train, Y_train)
Y_pred = model.predict(X_test)
compare_dataset = pd.DataFrame({'Actual': Y_test.flatten(),
                                'Predicted': Y_pred.flatten()})

print("\nFor SVM:")
print("\nConfusion Metrix:")
cm = metrics.confusion_matrix (Y_test, Y_pred)
print(cm)
print()
print("Accuracy: ", metrics.accuracy_score(Y_test, Y_pred))
print("Precision: ", metrics.precision_score(Y_test, Y_pred, average='macro'))
print("Recall: ", metrics.recall_score(Y_test, Y_pred, average='macro'))

y_pred_train = model.predict(X_train)

print("Training Accuracy: ", metrics.accuracy_score(Y_train, y_pred_train))
```

Output:

Confusion Matrix and Evaluation Metrics:

```
For SVM:

Confusion Matrix:
[[10  0  0]
 [ 0  8  1]
 [ 0  0 11]]

Accuracy:  0.9666666666666667
Precision:  0.9722222222222222
Recall:  0.9629629629629629
Training Accuracy:  0.9666666666666667
```

Model based on Logistic Regression:

```
#%%
print("\n\nFor Logistic Regression:")
from sklearn.linear_model import LogisticRegression

logistic_reg = LogisticRegression(C = 10, solver='newton-cg')

logistic_reg.fit(X_train, Y_train)
prediction_log_reg = logistic_reg.predict(X_test)

print("\nConfusion Matrix:")
cm = metrics.confusion_matrix (Y_test, prediction_log_reg)
print(cm)
print()

print("Accuracy: ", metrics.accuracy_score(Y_test, prediction_log_reg))
print("Precision: ", metrics.precision_score(Y_test, prediction_log_reg,average='macro'))
print("Recall: ", metrics.recall_score(Y_test, prediction_log_reg, average='macro'))

logistic_Y_pred_train = logistic_reg.predict(X_train)
print("Training Accuracy: ", metrics.accuracy_score(Y_train, logistic_Y_pred_train))
print()
```

Output:

Confusion Matrix and Evaluation Metrics:

```
For Logistic Regression:

Confusion Matrix:
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]

Accuracy:  1.0
Precision:  1.0
Recall:  1.0
Training Accuracy:  0.9916666666666667
```

Accuracy of the logistic regression-based model is 1 and the SVM based model is 0.9667. Since, the dataset has balanced number of classes so the classifier of logistic regression-based model is better than the classifier of SVM based-model which is indicating by the better accuracy number of logistic regression-based model.

ROC-Curve showing the AUC score:

Code:

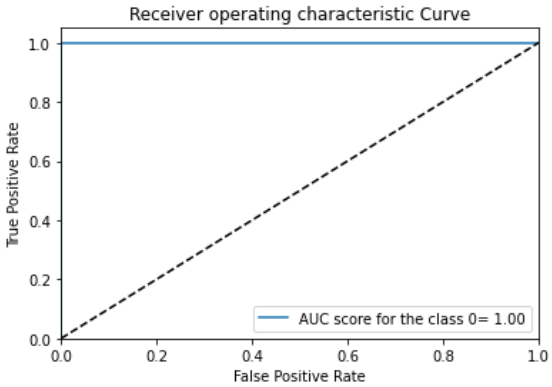
```
#%%

"""ROC Curve with AUC Score"""

from sklearn.metrics import roc_curve, auc
from sklearn.multiclass import OneVsRestClassifier
from sklearn import svm
from sklearn.preprocessing import label_binarize

classes=[0,1,2]
target = label_binarize(iris.target, classes=[0,1,2])
n_classes = 3

# shuffling and spliting training and test sets
X_train, X_test, y_train, y_test =train_test_split(iris.data, target, test_size=0.2, random_state=0)

# classifier
clf = OneVsRestClassifier(svm.SVC(kernel = 'linear', C = 10, probability = True))
y_score = clf.fit(X_train, y_train).decision_function(X_test)


# Computing ROC curve and ROC area for each class
fpr = dict()
tpr = dict()
roc_auc = dict()
for i in range(n_classes):
    fpr[i], tpr[i], _ = roc_curve(y_test[:, i], y_score[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])
```

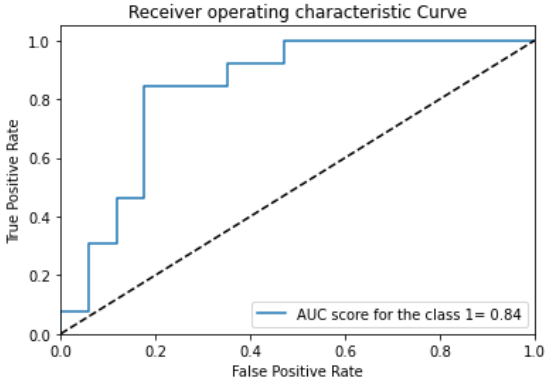
```
# Ploting ROC curve for a specific class

for i in range(n_classes):
    plt.figure()
    plt.plot(fpr[i], tpr[i], label='AUC score for the class '+str(classes[i])+'= %0.2f' % roc_auc[i])
    plt.plot([0, 1], [0, 1], 'k--')
    plt.xlim([0.0, 1.0])
    plt.ylim([0.0, 1.05])
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('Receiver operating characteristic example')
    plt.legend(loc="Lower right")
    plt.show()
```

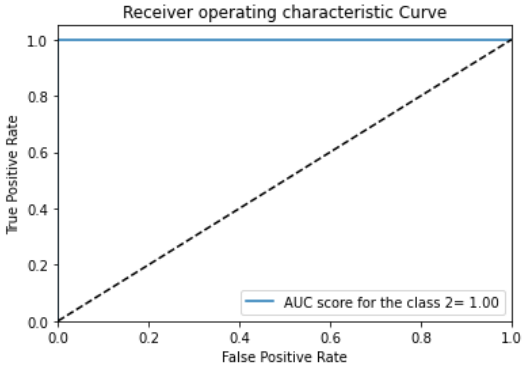
Here, the support vector machine method is used for creating the OneVsRestClassifier model.

Curves:

There are three classes in the iris dataset. So, there will be three curves.







Here, the ROC curves for class 0 and 2 are joined to the top left corner but the ROC curve for class 1 is joined to the lower part of the left side which is y-axis referring true positive rate. So, the classifiers for the model performed better to identify class 0 and 2. We can also see this from the AUC score for class 0 and 2 that the scores for those is higher than the AUC score of class 1. Which indicates the model performed better with the classifiers for classifying the class 0 and 2.

2. Learning outcomes:

From this lab, I have learned how to create and implement the logistic regression-based model and support vector machine-based model. I have learned how to evaluate the model with confusion matrix and evaluation metrices. I have also learned about the evaluation of the performance of models for identifying the classes in dataset by seeing the ROC curve and AUC scores.