Assignment 2: Scikit-learn

Francesco Penasa

December 12, 2019

1 Introduction

The second assignment of the Machine Learning require us to familiarize with the programming language Python3 and its most used libraries in the machine learning such as numpy, scipy, matplotlib and scikit-learn. For this assignment we have choosen the OCR a dataset among the three available. As classifier for the OCR problem we used support vector machine (SVM), at first we trained and tested the classifier with cross-validation of the training set and them we trained the classifier on the whole training set and tested it on the test set. In the next sections we will see some pieces of the Python3 code that has been used and the results emerged from the testing.

2 Test SVC with Cross-Validation

At first we imported the data and the labels, then we adjusted the data format to fit in the functions `cross_val_score()`. Then we search for the best gamma value between three looking at the one giving us the best accuracy. At the end after initializing the SVC with `C=10` and the `best_gamma` value, we evaluated the classifier with the training set using 3-cross-validation, below are displayed the evaluation metrics obtained and the core of the code used to obtain them.

```
# import
train_data = pd.read_csv('ocr/train-data.csv')
train_target = pd.read_csv('ocr/train-targets.csv')

# format the data
X_train, y_train = train_data, train_target
y_train = np.squeeze(np.asarray(y_train))

# find the best gamma
kf = KFold(n_splits = 3, shuffle = True, random_state = 42)
gammas = [0.4, 0.5, 0.6]
best_accuracy = best_gamma = 0
for gamma in gammas:
  clf = SVC(C=10, kernel='rbf', gamma=gamma)
  s_a = cross_val_score(clf, X_train, y_train, cv=kf, scoring='accuracy')
  if (s_a.mean() > best_accuracy):
    best_accuracy = s_a.mean()
    best_gamma = gamma

# evaluate the classifier with 3-Cross-Validation
clf = SVC(C=10, kernel='rbf', gamma=best_gamma)
s_p = cross_val_score(clf, X_train, y_train, cv=kf, scoring='precision_weighted')
s_r = cross_val_score(clf, X_train, y_train, cv=kf, scoring='recall_weighted')
s_f1 = cross_val_score(clf, X_train, y_train, cv=kf, scoring='f1_weighted')
```

Table 1: Evaluation metrics with 3-Cross-Validation.

accuracy	**precision_weighted**	**recall_weighted**	**f1_weighted**
0.89593	0.89682	0.89593	0.89539

3 Learning Curve

```
# import the train set
train_data = pd.read_csv('ocr/train-data.csv')
train_target = pd.read_csv('ocr/train-targets.csv')

# format the data
X_train, y_train = train_data, train_target
X_train = np.array(X_train)
y_train = np.squeeze(np.asarray(y_train))

# init the classifier and create learning curve
clf = SVC(C=10, kernel='rbf', gamma=best_gamma)
train_sizes, train_scores, val_scores =
  learning_curve(clf, X_train, y_train, scoring='accuracy', cv=3)
```

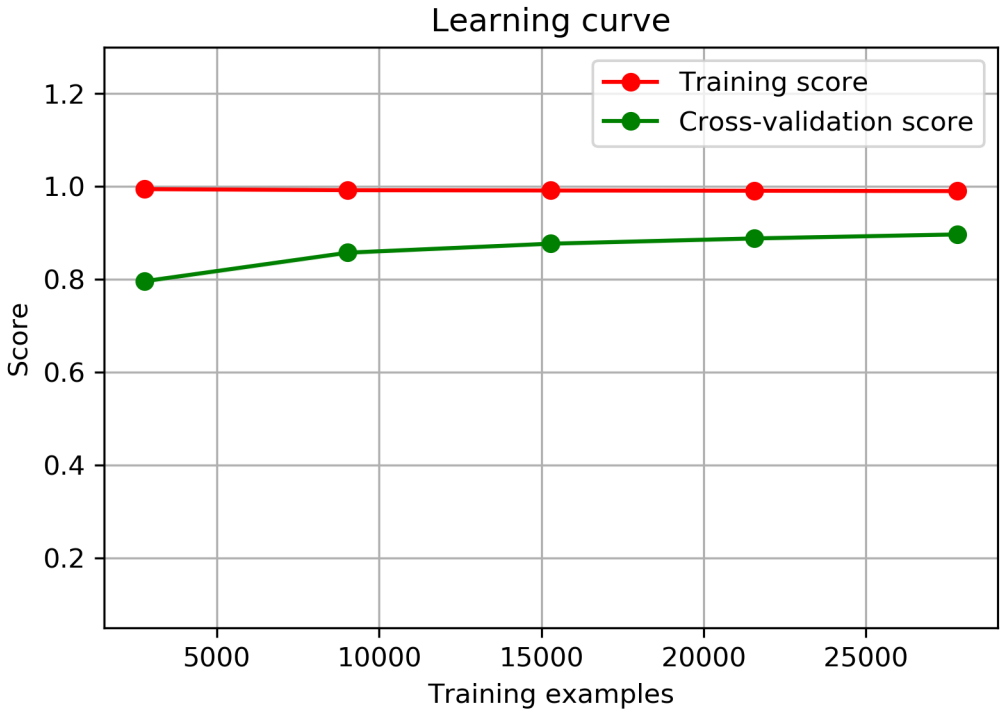

Figure 1: Learning Curve

4 Test SVC without Cross-Validation

At first we imported the data and the labels of both training and test set, then we adjusted the data format to fit the function `clf.fit()`. At the end after initializing the SVC with `C=10` and `gamma=best_gamma`, we trained the classifier with the whole training set and we predicted the labels of the test set using the test set data. below are displayed the evaluation metrics obtained and the core of the code used to obtain them.

```
# import
test_data = pd.read_csv('ocr/test-data.csv')
test_target = pd.read_csv('ocr/test-targets.csv')
train_data = pd.read_csv('ocr/train-data.csv')
train_target = pd.read_csv('ocr/train-targets.csv')

# format the data
X_train, y_train = train_data, train_target
X_test, y_test = test_data, test_target
y_train = np.squeeze(np.asarray(y_train))

# import the classificator
clf = SVC(C=10, kernel='rbf', gamma=best_gamma)

# training
clf.fit(X_train, y_train)

# prediction
y_pred = clf.predict(X_test)

# print the evaluation metrics
report = metrics.classification_report(y_test, y_pred)
print(report)
```

Table 2: Evaluation metrics without Cross-Validation.

accuracy	**precision_weighted**	**recall_weighted**	**f1_weighted**
0.90767	0.91	0.91	0.91

5 Conclusion

Since the threshold on the accuracy required to deliver this assignment was `0.09673` and we managed to reach an accuracy on the test set of `0.90767`, we are able to say that the SVM represents a good learning model for the OCR problem.