CS302 Introduction to machine learning

School of Undergraduate Studies, DGIST

201611161 정원균

Homework report 2.

2022. 05. 08

Q1. For MNIST data set, train Logistic regression models and find the best model that can achieve the highest accuracy on the test data set.

```
from sklearn import datasets, svm
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
import matplotlib.pyplot as plt
import numpy as np

X_digits, y_digits = datasets.load_digits(return_X_y=True)
X_digits = X_digits / X_digits.max()

num_sample = len(X_digits)

X_train = X_digits[:int(.9 * num_sample)]
y_train = y_digits[:int(.9 * num_sample)]
X_test = X_digits[int(.9 * num_sample):]
y_test = y_digits[int(.9 * num_sample):]
```

```
def linear_Reg():

    x = np.linspace(1,100,100)
    y = np.array([])
    for i in x:
        model = LogisticRegression(C = i, max_iter=1000)
        model_accuracy = model.fit(X_train, y_train).score(X_test, y_test)
        y = np.append(y, model_accuracy)

    plt.plot(x,y,'ro')
    max_acc = np.max(y)
    print("Max accuracy is: ", max_acc,"%" )

linear_Reg()
```

Figure 1. Library import and Linear regression code

```
Max accuracy is:  0.9444444444444444 %
```

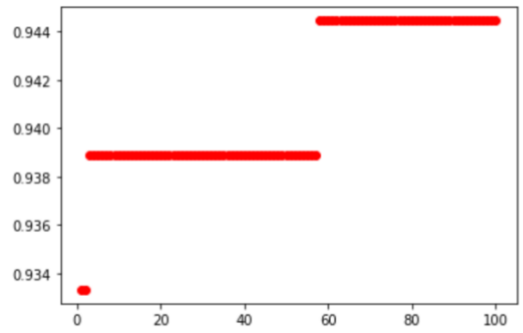

Figure 2. Accuracy of Linear Regression

Test 에 사용할 Logistic Regression, KNN, SVM, Random forest 모듈은 sklearn library 를 import 하여서 사용하였다. 또한 Mnist 데이터를 import 하여서 train data 와 test data 를 만들었다.

다음 linear_Reg( ) parameter 에서 LogisticRegression(C, max_iter)에서 규제강도를 의미하는 C 값을 0 부터 100 까지 나누어서 (x-label) Accuracy 를 분석해보았다. 결과를 보면 C 값이 60 이상일 때 94.4%정도의 정확도를 보임을 알 수 있다.

Q2. For the same data set, train K-NN classifiers and find the best model that can achieve the highest accuracy on the test data set

```
def KNN():

    x = np.linspace(1,100,100)
    y = np.array([])

    for i in x:
        model = KNeighborsClassifier(n_neighbors=int(i))
        model_accuracy = model.fit(X_train, y_train).score(X_test, y_test)
        y = np.append(y, model_accuracy)

    plt.plot(x,y,'ro')
    max_acc = np.max(y)

    print("Max accuracy is: ", max_acc,"%" )

KNN()
```

```
Max accuracy is:  0.9777777777777777 %
```

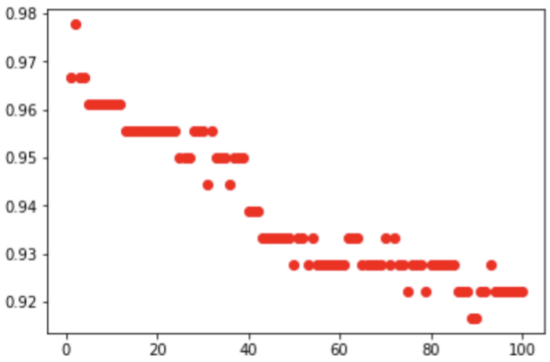

Figure 3. Code and Result of KNN

다음은 같은 Linear Regression 에 사용하였던 같은 train, test data 를 이용해서 KNN 모델에 적용한 코드와 결과이다. 앞에서 규제강도에 따라 Accuracy 를 확인했던 것과 달리 KNN model 에는 몇 개의 neighbor 을 비교할 것인지에 따라서 Accuracy 를 분석해보았다. 그 결과 K = 2 일때 가장 높은 정확도를 보였으며, K=2 가 아닌 case 에 대해서 비교할 Neighbor 의 수가 증가할 수록 오히려 정확도가 떨어지는 모습을 볼 수 있다.

Q3. For the same data set, train SVM classifiers and find the best model that can achieve the highest accuracy on the test data set.

```
def SVM():

    x = np.linspace(1,100,100)
    y = np.array([])
    n = 1
    kernel = ['linear','poly','rbf','sigmoid']

    plt.figure(figsize = (15,10))

    for ker in kernel:
        for i in x:
            model = svm.SVC(gamma = 0.01, C = i,kernel = ker)
            model_accuracy = model.fit(X_train, y_train).score(X_test, y_test)
            y = np.append(y, model_accuracy)

        max_acc = np.max(y)

        print("SVM Kernel with",ker,"accuracy:",max_acc)

        plt.subplot(2, 2, n)
        plt.plot(x,y,'ro')
        y = np.array([])
        n+=1

SVM()
```

Figure 4. Code of SVM model

```
SVM Kernel with linear accuracy: 0.9666666666666667
SVM Kernel with poly accuracy: 0.9333333333333333
SVM Kernel with rbf accuracy: 0.9666666666666667
SVM Kernel with sigmoid accuracy: 0.9611111111111111
```

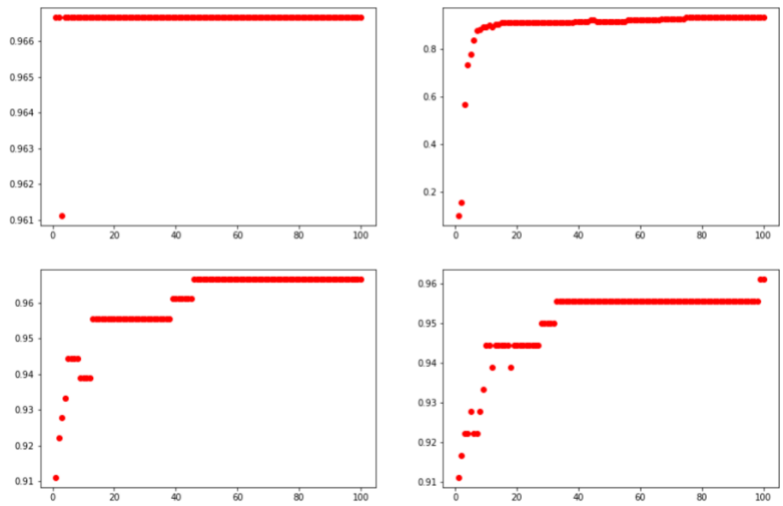

Figure 5. Result of SVM model

SVM에서는 총 두가지의 parameter의 변화를 주어서 Accuracy를 확인해보았다. Kernel은 총 4개 linear, polynomial, rbf, sigmoid의 결과를 확인하였고, 역시 각 kernel별로 규제정도 C를 1부터 100까지 바꾸어 가면서 각 kernel 별로 가장 높은 Accuracy를 출력하였다. 그 결과 linear model과 rbf model이 96.7% 정도로 가장 높음을 알 수 있고 C값은 50이상의 수를 parameter로 주면 문제없이 가장 높은 Accuracy가 나온다는 것을 알 수 있다.

Q4. For the same data set, train Random forest classifiers and find the best model that can achieve the highest accuracy on the test data set.

```
def Random_forest():

    x = np.linspace(100,1000,100)
    y = np.array([])

    for i in x:
        model = RandomForestClassifier(n_estimators = int(i), max_depth=20,random_state=0)
        model_accuracy = model.fit(X_train, y_train).score(X_test, y_test)
        y = np.append(y, model_accuracy)

        max_acc = np.max(y)

    plt.plot(x,y,'ro')
    print("Max accuracy is: ", max_acc,"%" )


Random_forest()
```

```
Max accuracy is:  0.9388888888888889 %
```

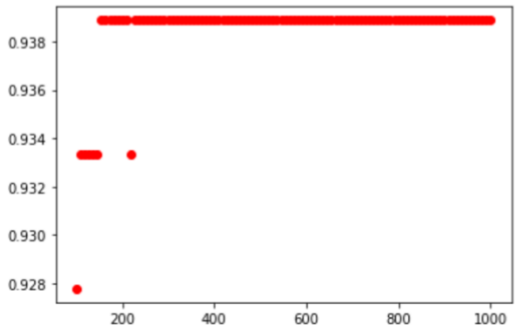

Figure 6. Code & Result of Random Forest model

다음은 max depth = 20, Estimator 수를 100 ~ 1000 까지 준 Random forest model 학습 결과이다. Estimator 를 약 180 개 이상 대입하였을 때 약 93.9% 의 정확도를 보임을 알 수 있다.

Conclusion

	Linear Reg.	KNN	SVM	Random Forest
Accuracy	94.4%	97.8%	96.7%	93.3%

총 4 개 유형의 model 을 바탕으로 각 model 당 변화시킬 수 있는 parameter 를 바꾸면서 최대 정확도를 구해보았을 때 KNN model 에서 n of neighbor 가 2 일 때 가장 높은 정확도를 보여준다는 것을 알 수 있다.