빅데이터분석

김희찬 교수님

MNIST 숫자 데이터를 이용한 분류 모델 학습 및 평가

소프트웨어학부 20163231 신승은

#############데이터 전처리, 분할 등 실험 전 과정을 정리###############

자주 반복해서 쓰는 부분들을 Util.py에 get10fold함수, runCV함수, load\_mnist함수를 작성하여 실험 파일에서 호출합니다.

Get10fold함수는 데이터를 10개로 잘라 train을 할 데이터와 아닌 함수를 나누는 함수

Get\_one\_fold함수는 get10fold함수를 일반화하여 모든 fold수에 대응하는 함수

Load\_mnist함수는 MNIST 데이터의 csv 파일을 읽어와 정규화된 데이터를 호출하는 모듈과 함수

runCV1함수는 숫자 데이터를 이용하여 성능을 검증하는 함수

runCV2함수는 precision,recall을 구하기 위한 함수

<Decision Tree> #0.85

학습데이터와 레이블을 load\_mnist함수로 불러온다.

Random.shuffle()로 주어진 리스트형 데이터의 인덱스를 무작위로 섞는다.

DecisionTreeClassifier() 객체를 생성하고, fit함수로 데이터와 레이블을 모델학습한다.

Shuffled한 인덱스를 넣은 데이터와 레이블을 runCV1함수로 성능값을 도출한다.

Np.mean함수로 성능의 평균값을 출력한다.

<naïve bayes> #0.55

학습데이터와 레이블을 load\_mnist함수로 불러온다.

Random.shuffle()로 주어진 리스트형 데이터의 인덱스를 무작위로 섞는다.

GaussianNB() 객체를 생성하고, fit함수로 데이터와 레이블을 모델학습한다.

Shuffled한 인덱스를 넣은 데이터와 레이블을 runCV1함수로 성능값을 도출한다.

Np.mean함수로 성능의 평균값을 출력한다.

<KNN> ##너무 오래걸려서 결과물을 확인 못함

학습데이터와 레이블을 load\_mnist함수로 불러온다.

Random.shuffle()로 주어진 리스트형 데이터의 인덱스를 무작위로 섞는다.

KNeighborsClassifier() 객체를 생성하고, fit함수로 데이터와 레이블을 모델학습한다.

Shuffled한 인덱스를 넣은 데이터와 레이블을 runCV1함수로 성능값을 도출한다.

Np.mean함수로 성능의 평균값을 출력한다.

<logistic regression> #105.68

학습데이터와 레이블을 load\_mnist함수로 불러온다.

Random.shuffle()로 주어진 리스트형 데이터의 인덱스를 무작위로 섞는다.

LogisticRegression() 객체를 생성한다.

Shuffled한 인덱스를 넣은 데이터와 레이블을 runCV2함수로 결과를 도출한다.

<perceptron> #40.15

학습데이터와 레이블을 load\_mnist함수로 불러온다.

Random.shuffle()로 주어진 리스트형 데이터의 인덱스를 무작위로 섞는다.

perceptron(max\_iter=500,n\_jobs=3) 객체를 생성한다. (max\_iter은 최대 epoch 횟수, n\_jobs는 멀티 프로세스의 개수)

Shuffled한 인덱스를 넣은 데이터와 레이블을 runCV2함수로 결과를 도출한다.

Np.mean함수로 결과의 평균값을 출력한다.

\*\*학습률 eta0=0.1로 설정할 경우 0.775의 성능 결과가 나온다.

<MLP> #2126.86

학습데이터와 레이블을 load\_mnist함수로 불러온다.

Random.shuffle()로 주어진 리스트형 데이터의 인덱스를 무작위로 섞는다.

Mlp(hidden\_layer\_sizes=20,max\_iter=500)객체를 생성한다. (hidden\_layer\_sizes는 은닉층의 수, max\_iter은 최대 epoch수)

Shuffled한 인덱스를 넣은 데이터와 레이블을 runCV2함수로 결과를 도출한다.

Np.mean함수로 결과의 평균값을 출력한다.

\*\*학습률 learning\_rate\_init=0에서 0.1로 올렸을 때의 성능이 0.85에서 0.76으로 낮아진다. 하지만 안좋다고 말할 순 없다.

<randomforest> #0.96

학습데이터와 레이블을 load\_mnist함수로 불러온다.

Random.shuffle()로 주어진 리스트형 데이터의 인덱스를 무작위로 섞는다.

randomForestClassifier() 객체를 생성한다.

Shuffled한 인덱스를 넣은 데이터와 레이블을 runCV2함수로 결과를 도출한다.

Np.mean함수로 결과의 평균값을 출력한다.

<LDA> #0.86

학습데이터와 레이블을 load\_mnist함수로 불러온다.

Random.shuffle()로 주어진 리스트형 데이터의 인덱스를 무작위로 섞는다.

LinearDiscriminantAnalysis() 객체를 생성한다.

Shuffled한 인덱스를 넣은 데이터와 레이블을 runCV2함수로 결과를 도출한다.

Np.mean함수로 결과의 평균값을 출력한다.

<QDA> #강의에선 실행하지 않음->0.19

학습데이터와 레이블을 load\_mnist함수로 불러온다.

Random.shuffle()로 주어진 리스트형 데이터의 인덱스를 무작위로 섞는다.

QDA() 객체를 생성한다.

Shuffled한 인덱스를 넣은 데이터와 레이블을 runCV2함수로 결과를 도출한다.

Np.mean함수로 결과의 평균값을 출력한다.

<SVM> #0.91

학습데이터와 레이블을 load\_mnist함수로 불러온다.

Random.shuffle()로 주어진 리스트형 데이터의 인덱스를 무작위로 섞는다.

SVC() 객체를 생성한다.

Shuffled한 인덱스를 넣은 데이터와 레이블을 runCV2함수로 결과를 도출한다.

Np.mean함수로 결과의 평균값을 출력한다.

<GridSearch>

파라미터 공간을 일정구간으로 나누고 각 지점에 대하여 학습 및 평가를 진행한다.

학습데이터와 레이블을 load\_mnist함수로 불러온다.

Random.shuffle()로 주어진 리스트형 데이터의 인덱스를 무작위로 섞는다.

테스트할 파라미터 리스트 ‘linear’, ‘rbf’, ‘poly’, ‘sigmoid’ 를 생성한다.

파라미터별로 모델 생성하여 학습 및 평가를 한다.

Best\_param을 출력하면 결과물을 볼 수 있다