2018년 1학기 IT 집중교육 1

Homework 1

**[1] 데이터 분석과제**

Autouniv는 classification model을 만들기 위해 인위적으로 데이터를 만들어내는 모델이다. 우리의 목표는 Autouniv 모델을 통해 만들어진 ‘au4\_2500.csv’(2500 instances, 100 features) 데이터를 통해 분류기(KNN, Decision Tree, SVM)를 만들고 이를 최적화하여 높은 예측 정확도를 나타내도록 하는 것이다. 데이터는 80%의 training data(2000개, au4\_2500\_train.csv)와 20%의 test data(500개, au4\_2500\_test.csv)로 나누었다.

\* ipynb 파일을 제출하는 경우 주석을 이용하여 문제에 대해 간단하게 설명하고 코드에 대한 설명을 포함해야 한다.

\* 출처: https://sites.google.com/site/autouniv/autouniv-home

\* 참조: https://archive.ics.uci.edu/ml/datasets/AutoUniv

\* [1] 데이터분석 문제는 sklearn 패키지 이용 가능

\* 최종성능은 최소 50% 이상이어야한다.

\* 코드 및 분석 내용을 copy할 경우 0점 처리

\* 제출기한

(1), (2)은 3/11(일) 11:59 p.m. 까지 ipynb파일을 아주BB에 제출

(3)은 3/12(월) 수업시간 시작 전에 프린트해서 제출

1. **데이터 분석(1)**: au4\_2500\_train.csv 파일을 이용하여 분류기를 학습시킨 뒤, au4\_2500\_test.csv 의 label을 예측해보고 이를 실제 정답(y\_test)과 비교하여 예측 정확도를 구해본다.
2. 제공된 au4\_2500.csv 파일을 읽고 5개의 feature를 뽑아 EDA를 수행한다(이 과정에서 기술 통계량, 시각화(Plot)가 수행되어야 한다. 그리고 5개의 feature는 임의로 선택해도 되고 통계 수행을 통한 사전지식을 이용하여 선택해도 된다).
3. K 값을 5로 설정하고 훈련데이터(au4\_2500\_train.csv)를 이용해 5-NN 분류기를 학습한 뒤, 주어진 테스트 데이터(au4\_2500\_test.csv)의 class를 예측하고 이를 y\_test와 비교해 예측 정확도를 구한다.
4. 훈련데이터(au4\_2500\_train.csv)를 이용해 Decision tree 분류기를 학습한 뒤, 주어진 테스트 데이터(au4\_2500\_test.csv)의 class를 예측하고 이를 y\_test와 비교해 예측 정확도를 구한다.
5. 훈련데이터(au4\_2500\_train.csv)를 이용해 SVM 분류기를 학습한 뒤, 주어진 테스트 데이터(au4\_2500\_test.csv)의 class를 예측하고 이를 y\_test와 비교해 예측 정확도를 구한다..

* 위의 C, D, E의 분류기를 학습시킬 때 모든 feature를 사용할 필요는 없다.
* C, D, E 과정 모두 최종 training accuracy(training data에 대해)와 test accuracy(test data에 대해)를 각각 구해야 한다.

1. **데이터 분석(2)** 
   1. Cross-Validation을 이용하여 데이터 분석(2)에서 사용한 알고리즘의 성능을 높일 수 있도록 최적화 과정을 수행한다(e.g. 사용하는 feature의 개수, K-NN 분류기의 K, SVM 분류기의 gamma, hyper-parameter 설정, 검증 데이터를 이용하여 성능 판단).
   2. 2-A에서 찾은 최적 분류기의 검증 데이터 결과를 Confusion Matrix의 형태로 나타내고 Precision, Recall, F-1 Score를 구한다.
2. **보고서작성**: 데이터 분석(1), (2)를 수행한 결과를 종합하여 보고서를 작성한다(분량은 그림제외 5장 이내, 코드 포함하지 말 것).
   1. 이전에 수행했던 일들을 정리하여 보고서 작성(양식은 샘플 참조).
   2. 분류를 위해 사용한 여러 알고리즘의 성능을 비교하고 결과에 대한 해석 수행

**[2] KNN classifier 구현**

\* numpy package만을 이용하여 구현(sklearn 이용 불가)

\* mushrooms\_train(KNN).csv: training data, mushrooms\_test(KNN).csv: test data로 사용

\* 코드 copy할 경우 0점 처리

\* 제출기한: 3/11(일) 11:59 p.m. 까지 소스코드를 아주BB에 제출

1. KNN classifier class(KNN.py)를 생성
   1. train 함수를 만들어 KNN classifier를 학습시키는 기능 구현
   2. get\_distance 함수를 만들어 테스트 데이터와 학습된 데이터간의 거리를 구하는 기능 구현(여러 distance measure가 존재할 수 있음)
   3. predict 함수를 만들어 테스트 데이터의 정답(label)을 예측하는 기능 구현
2. Simulation.py를 통해 위에서 만든 KNN classifier로 classification 수행
   1. 제공된 데이터를 이용하여 K=5일 때 KNN을 수행
   2. 학습된 KNN classifier의 training accuracy 출력
   3. (정답 개수/테스트 데이터 개수) 로 최종 KNN classifier의 test accuracy 출력