2017-08

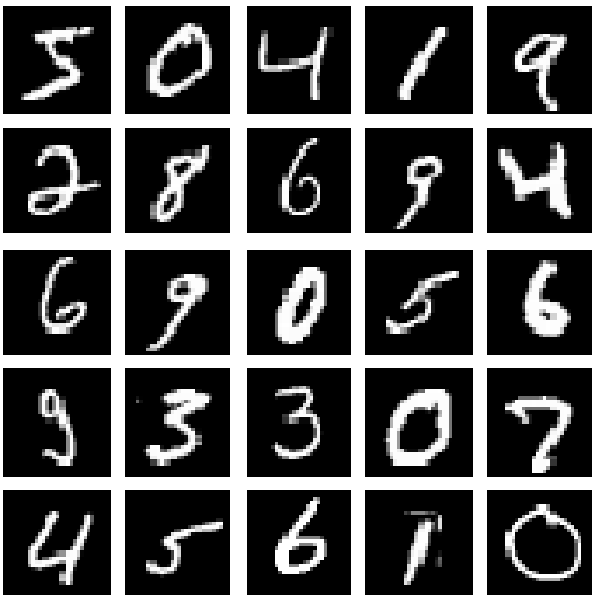
기계학습 실습

MNIST

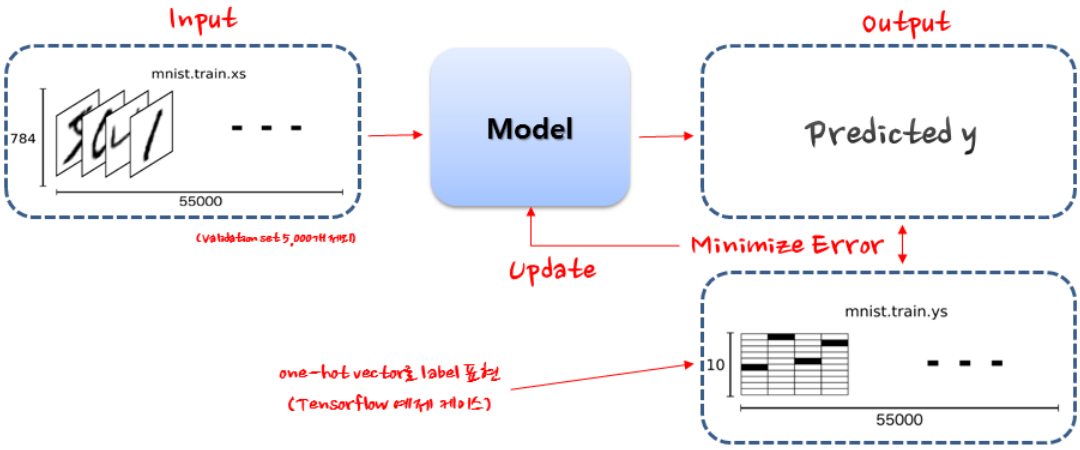
Weka 이용해서 MNIST 데이터 학습시키기

Index

1. 숫자 이미지 데이터 예측



위와 같이 손으로 쓰여진 0부터 9까지의 숫자를 스캔한 MNIST 데이터 셋을 이용하여 특정 모델을 학습시킨 후 임의의 숫자 이미지 데이터가 0~9 중 어느 숫자에 속하는지에 대한 값을 예측하게 된다.



1.1. 데이터 소개

1.1.1. MNIST란?

Mixed National Institute of Standards and Technology database의 약어로, Yann LeCun 교수가 제공한다. 이 데이터 각각은 784(28*28 pixel)개의 값으로 각 값은 0~255사이의 값을 가진다. 각 데이터는 어떤 숫자를 의미하는지에 대한 값을 가지고 있다. 데이터는 총 70000개로, 60000개의 training set, 10000개의 test set으로 구성되어 있다.

1.2. Previous experiments

1. http://yann.lecun.com/exdb/mnist/
2. http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html

1.3. MNIST arff DataSet 다운로드

https://github.com/diana-jinhyeon/note

```
@attribute pixel776 numeric
@attribute pixel777 numeric
@attribute pixel778 numeric
@attribute pixel779 numeric
@attribute pixel780 numeric
@attribute pixel781 numeric
@attribute pixel782 numeric
@attribute pixel783 numeric
@attribute label {0,1,2,3,4,5,6,7,8,9}

@data
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,18,30,137,137,192,
86,72,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13,86,250,254,254,254,254,217,246,151,3
2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,16,179,254,254,254,254,254,254,254,254,254,231,5
4,15,0,0,0,0,0,0,0,0,0,0,0,0,0,0,72,254,254,254,254,254,254,254,254,254,254,254,25
4,104,0,0,0,0,0,0,0,0,0,0,0,0,0,0,61,191,254,254,254,254,254,109,83,199,254,254,254,
254,243,85,0,0,0,0,0,0,0,0,0,0,0,0,172,254,254,254,202,147,147,45,0,11,29,200,254,
254,254,171,0,0,0,0,0,0,0,0,0,0,0,1,174,254,254,89,67,0,0,0,0,0,0,128,252,254,254,2
12,76,0,0,0,0,0,0,0,0,0,0,47,254,254,254,29,0,0,0,0,0,0,0,0,0,83,254,254,254,153,0,0,0
,0,0,0,0,0,0,0,80,254,254,240,24,0,0,0,0,0,0,0,0,0,25,240,254,254,153,0,0,0,0,0,0,0,0,
0,0,64,254,254,186,7,0,0,0,0,0,0,0,0,0,166,254,254,224,12,0,0,0,0,0,0,0,0,0,14,232,25
4,254,254,29,0,0,0,0,0,0,0,0,0,75,254,254,254,17,0,0,0,0,0,0,0,0,18,254,254,254,254
,29,0,0,0,0,0,0,0,0,0,48,254,254,254,17,0,0,0,0,0,0,0,0,2,163,254,254,254,29,0,0,0,0
,0,0,0,0,0,48,254,254,254,17,0,0,0,0,0,0,0,0,0,94,254,254,254,200,12,0,0,0,0,0,0,0,1
6,209,254,254,150,1,0,0,0,0,0,0,0,0,0,15,206,254,254,254,202,66,0,0,0,0,0,21,161,2
54,254,245,31,0,0,0,0,0,0,0,0,0,0,0,60,212,254,254,254,194,48,48,34,41,48,209,254,
254,254,171,0,0,0,0,0,0,0,0,0,0,0,0,0,86,243,254,254,254,254,254,233,243,254,254,2
54,254,254,86,0,0,0,0,0,0,0,0,0,0,0,0,0,0,114,254,254,254,254,254,254,254,254,254,
254,239,86,11,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13,182,254,254,254,254,254,254,254,254,2
43,70,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,8,76,146,254,255,254,255,146,19,15,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
```

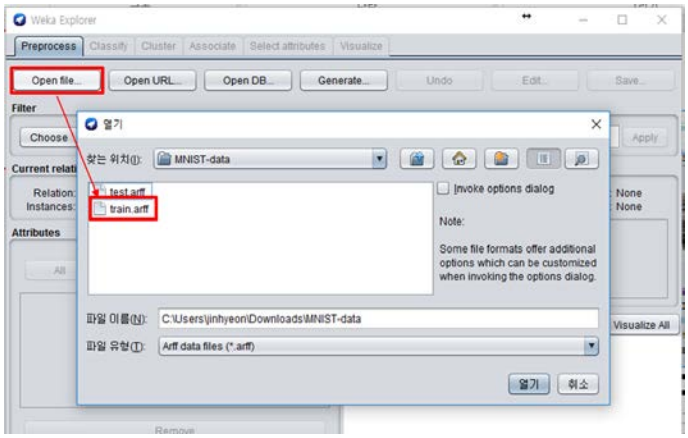
2. Weka로 데이터 전처리 하기

MNIST data의 사이즈가 WEKA의 heap size를 넘기 때문에 command-line에서 java -Xmx1024m -jar weka.jar 입력하여 Weka를 불러온다. 1024는 시스템에 맞추어 적절히 정한다.

2.1. MNIST Dataset 불러오기

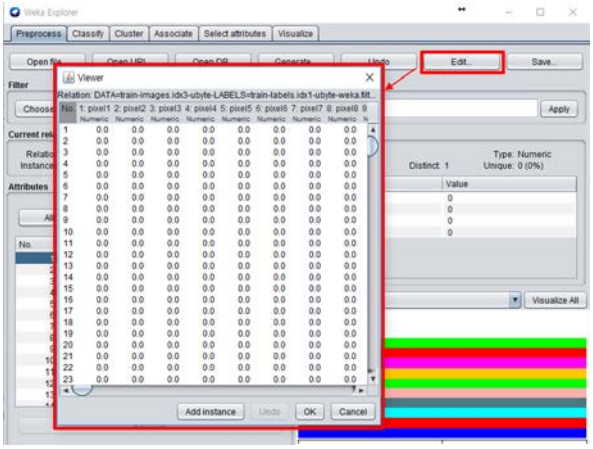
① Explorer 선택

② Open file > 파일 유형 – csv > 다운받은 train.csv > 열기 선택



2.2. Data 확인

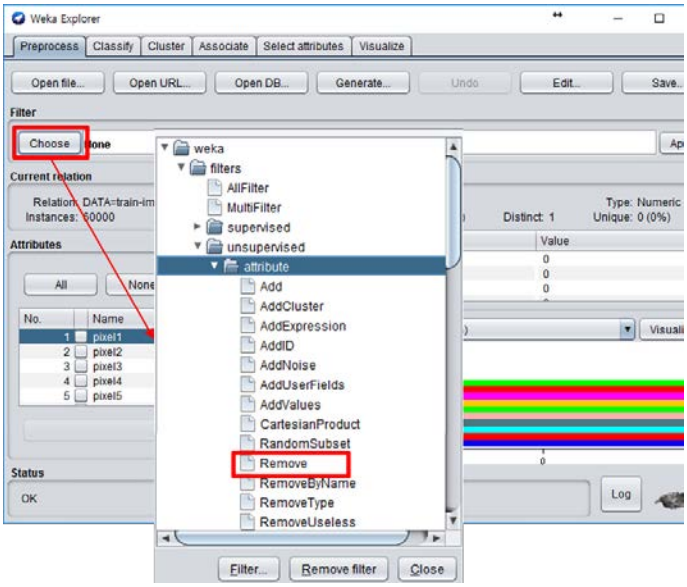
① Edit > Viewer



2.3. Data Handling

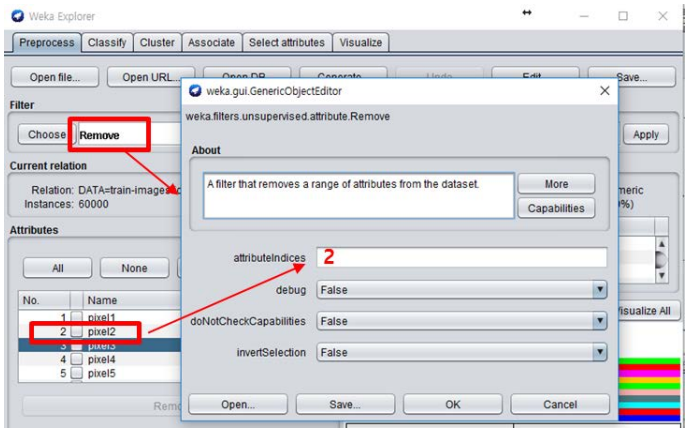
2.3.1. Remove : attribute를 선택하여 삭제

① Filter > weka > filters > unsupervised > attribute



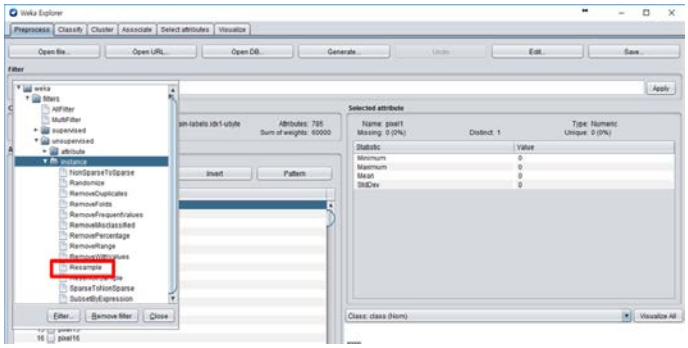
참고) Filter > supervised는 classifier를 선택하여 classifier 기준으로 data를 filtering할 수 있도록 하나, 여기서는 사용자가 원하는 특정 attribute의 no를 호출하여 삭제하는 것을 원하므로 unsupervised 메뉴에서 선택한다.

② Remove > attributeIndicate에 삭제하려는 attribute의 No. 입력 > OK > Apply

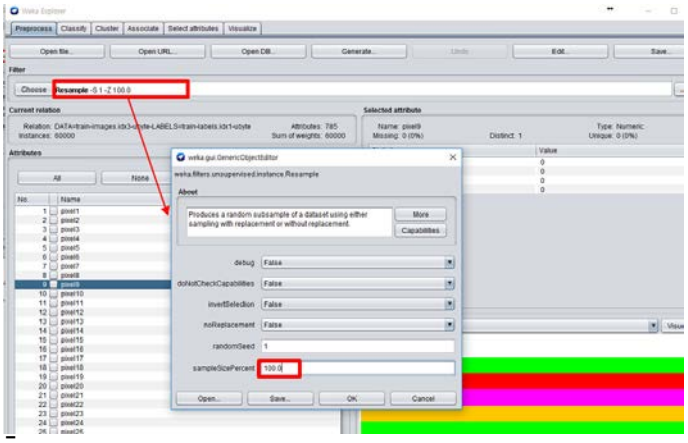


2.3.2. Resampling

① Filter > Choose > Weka > filters > unsupervised > instance > Resample



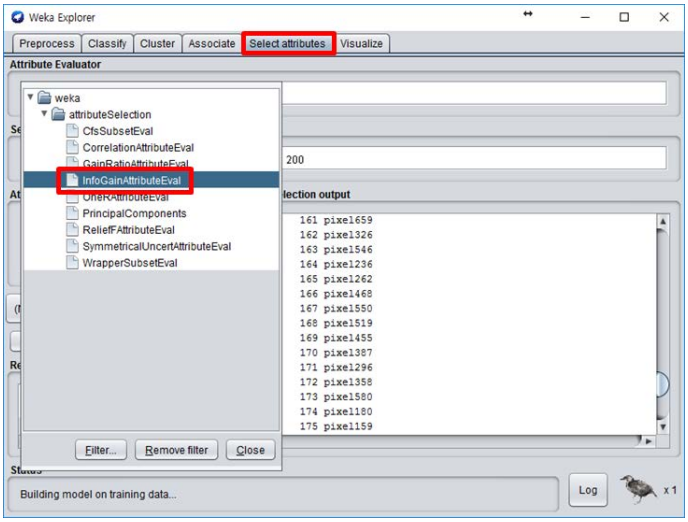
② Resample 클릭 > samplingSizePercent > OK > Apply



참고) samplingSIzePercent에 전체 데이터 사이즈 중 사용할 데이터의 퍼센트를 입력한다.

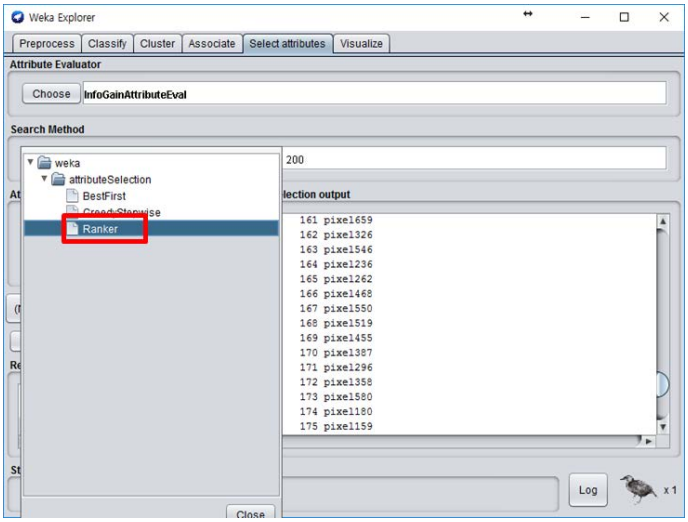
2.3.3. Select Attribute

① Select attributes > Attribute Evaluator – choose > InfoGainAttributeEval



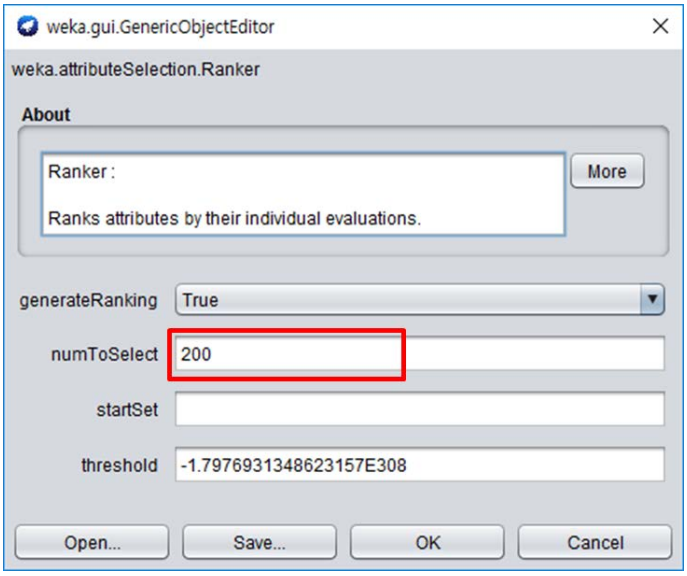
참고) InfoGain은 gain값을 기준으로 attribute를 선택하는 매서드이다.

② Search Method – choose > Ranker



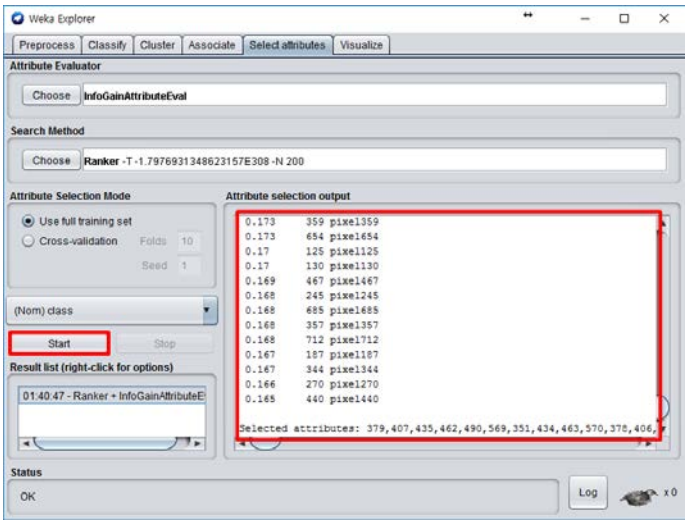
참고) InfoGain은 attribute를 rank하여 원하는 개수를 선택하는 매서드이다. Ranker를 사용한다.

③ Ranker > numToSelect



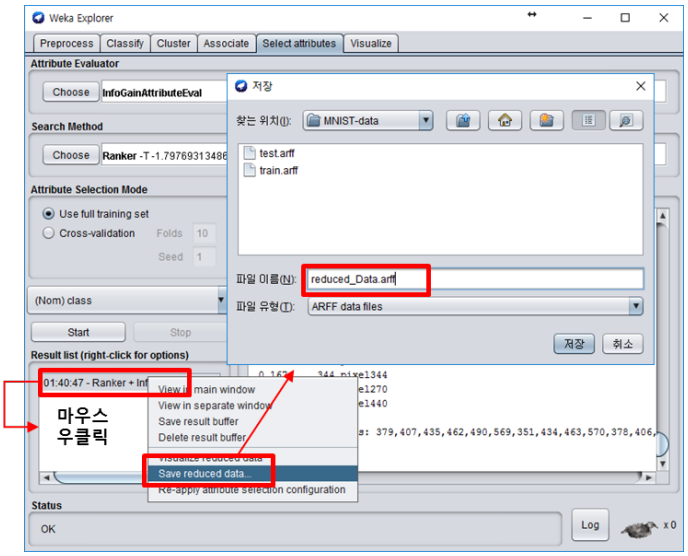
참고) Ranker의 numToSelect는 attribute를 gain값 순서대로 rank를 매겨서 선택할 attribute의 갯수이다.

④ Start > 값 확인

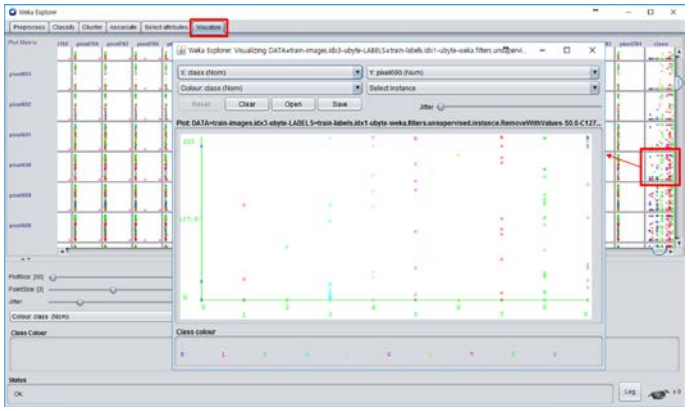


참고) gain 순서대로 rank되어 200개의 선택된 attribute들을 확인할 수 있다.

⑤ Result list > 결과 마우스 우클릭 > save reduced data > 이름 설정 후 저장



2.4. MNIST Dataset Visualize



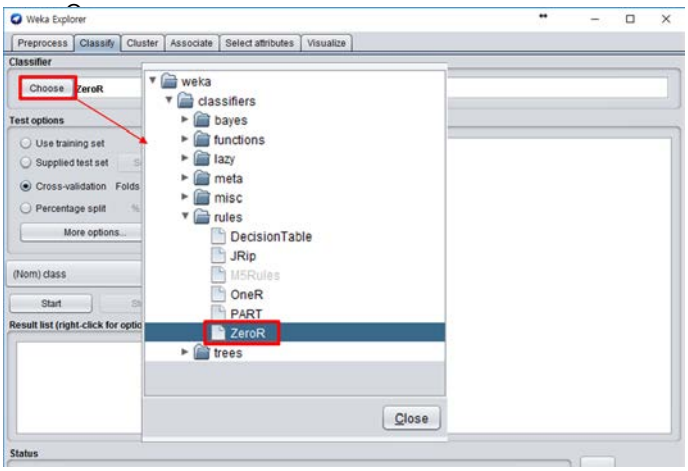
참고) 모든 attribute의 조합을 보여주며 각 칸은 두개의 attribute를 이용하여 2차원으로 보여주는 형태이다. Class별 색상이 정해져 있다.

3. Weka로 기계학습 모델 학습시키고 예측하기

3.1. Base Line (Zero R, One R)

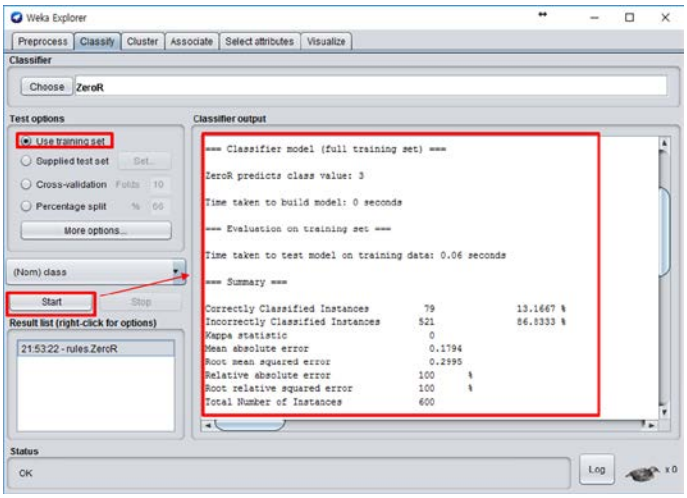
3.1.1. Zero R : 모든 값을 하나의 클래스로 예측

① Classify > Choose > weka > classifiers > rules > ZeroR



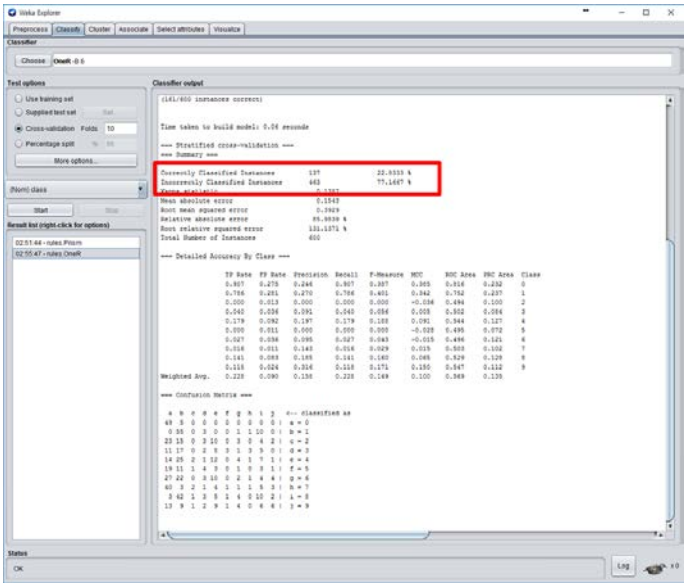
참고) Zero R은 Baseline으로 쓰이는 알고리즘이다. 모든 attribute를 class중 가장 많은 class로 예측한다. Baseline으로 사용되기 때문에 test option역시 use training set으로 설정한다.

② Use training Set > Start



참고) Zero R classifier는 MNIST데이터의 attribute들을 모두 3으로 분류되는 class로 예측하였다. 0~9까지의 class 중 13%가 3이기 때문에 13%의 데이터를 옳게 분류하였다.

3.1.2. One R : 하나의 attribute를 기준으로 class 분류

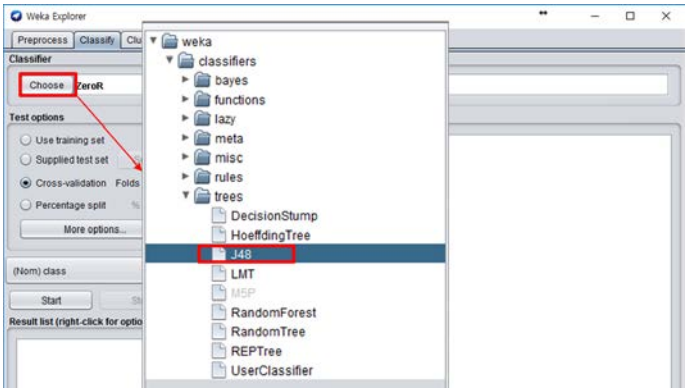


참고) 에러율 확인해서 가장 에러율이 작은 하나의 attribute를 골라서 class를 나누는 classifier이다.

아주 간단한 데이터셋이나 noise가 많이 포함된 데이터셋, 데이터에서 학습할 것이 없을 때 사용한다

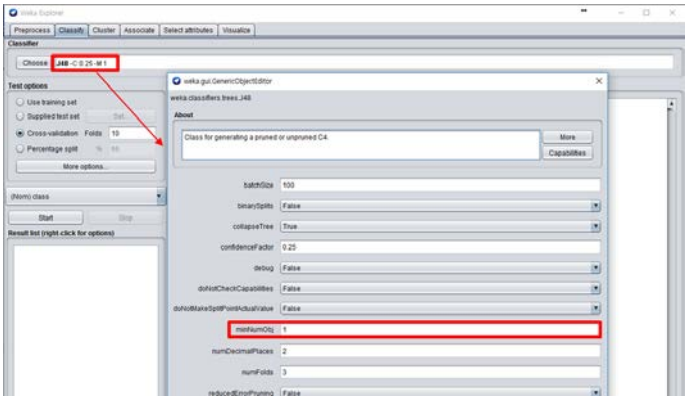
3.2. Tree (J48 : decision tree algorithm.)

① Classify > Choose > weka > classifiers > trees > J48



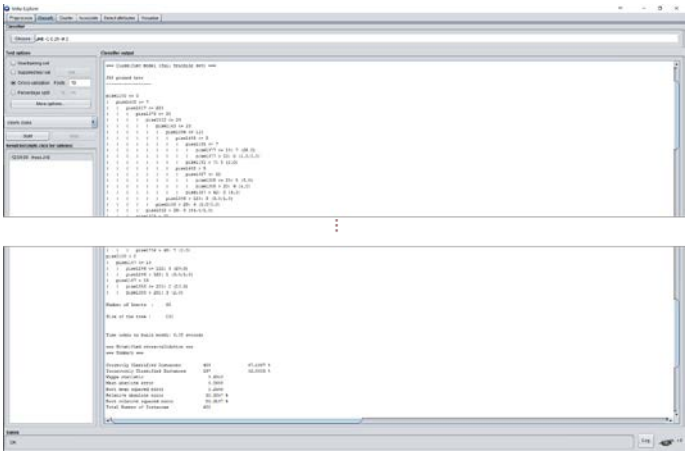
참고) Ross Quinla이 ID3를 기반으로 구현한 C45 알고리즘의 개정판 C48을 weka에서 java로 구현하여 J48이라는 이름을 붙였다. continuous하거나 discrete한 attribute를 처리할 수 있고, missing value를 처리하여 모델을 학습시킬 수 있으며 각 attribute에 다른 cost를 줄 수 있고 pruning도 가능하다.

② (선택사항) J48 > minNumObj의 숫자를 정한다.



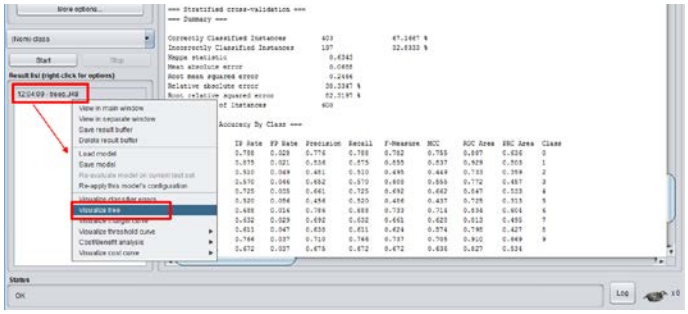
참고) minNumObj말고도 다양한 조건을 추가할 수 있다. Tree를 Prun하기위한 값 조정도 가능하다.

③ Cross-validation > start

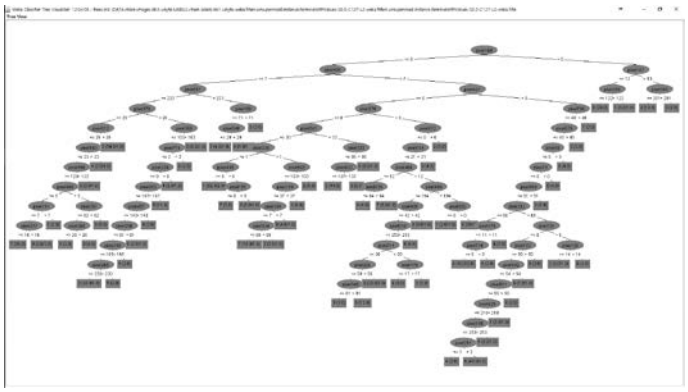


참고) 모델의 generalization을 위하여 Cross validation을 시행하였다.
Decision Tree는 67.1667 % 의 정확도를 보인다. 위의 그래프는 트리를 text로 시각화 한 것이다.

④ 결과 목록에서 우클릭 > Visualize Tree

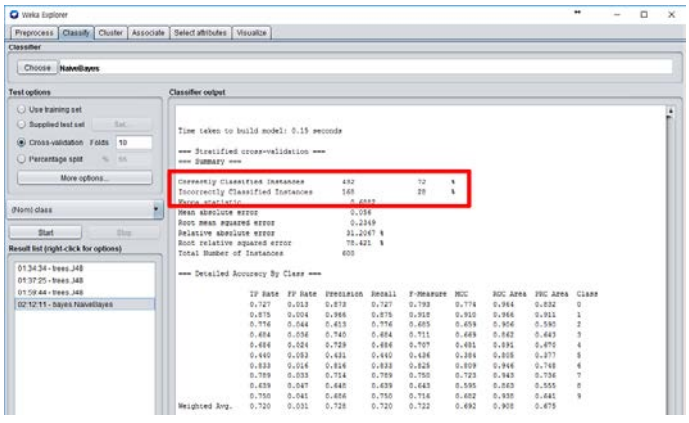


⑤ Tree 확인하기



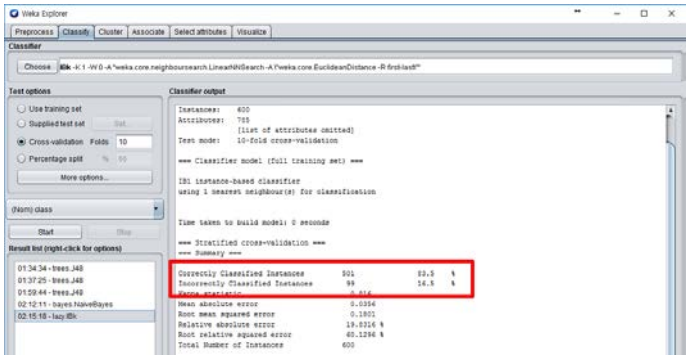
참고) tree를 visualize했을 시, 볼 수 없을 만큼 작게 나오는 경우가 있다. 윈도우의 크기를 키운 후, 오른쪽 버튼 클릭하여 fit to Screen을 클릭한다.
원형의 노드는 기준을 제시하고 연결선은 참인지 거짓인지 판별한다. 사각형의 노드는 분류된 값이다.

3.3. Bayse(Naïve bayes)



3.4. Lazy(IBk : K-nearest neighbor classifier)

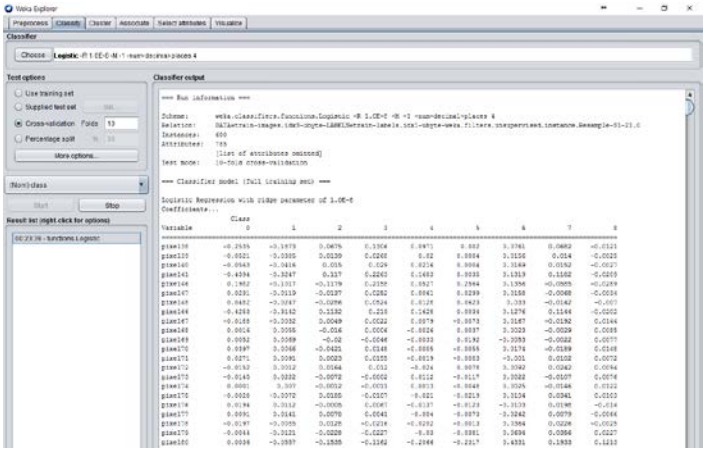
① Lazy > IBk



3.5. Linear Classifier (Logistic, Perceptron)

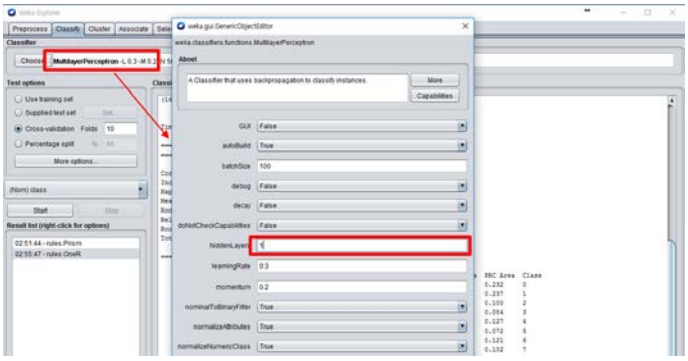
3.5.1. Logistic

① function > Logistic



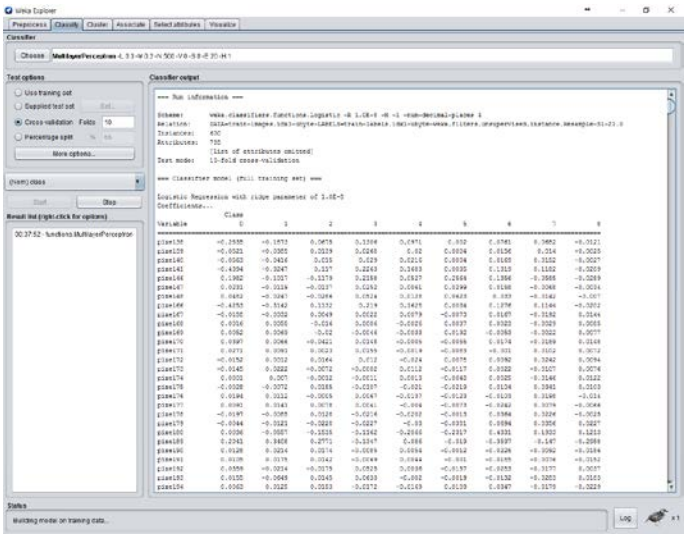
3.5.2. Perceptron

① Function > multilayerPerceptron



참고) Multi-Layer Perceptron의 hidden layer 개수를 조정하여 perceptron을 만든다. 1은 single perceptron이고, a 는 hidden layer를 auto로 설정하는 것이다.
상당히 긴 시간이 걸린다.

② 상당히 오랜 시간이 걸려서 결과가 출력되는 것을 확인할 수 있다.

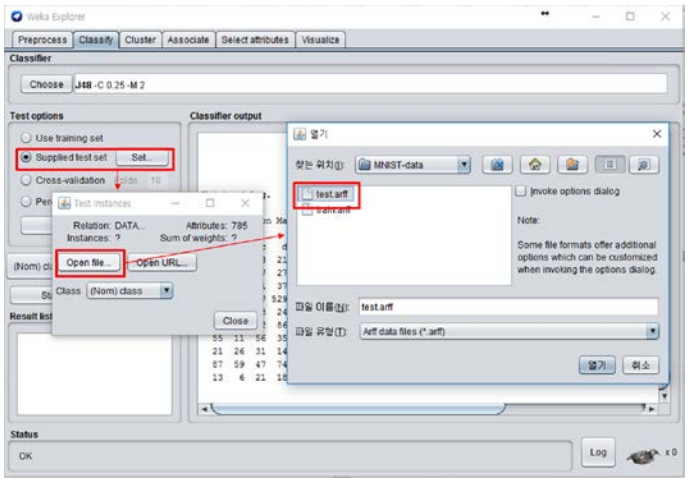


3.6. Evaluation

3.6.1. Test set

① Traning set은 앞에서 실행한 Preprocess 탭에서 데이터 파일을 여는 형식으로 불러온다. 우리는 MNIST training set을 불러와 resampling 해 두었다..

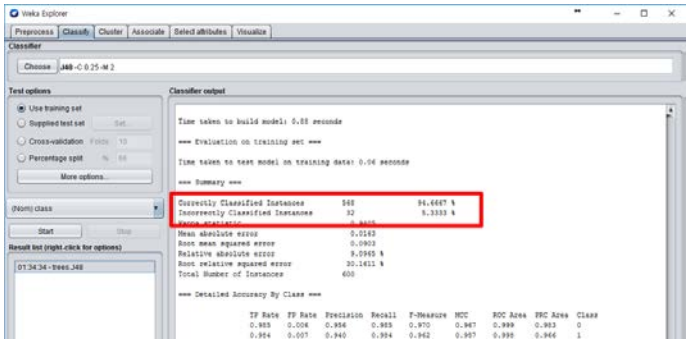
② Test set은 아래와 같이 설정한다.



참고) MNIST는 test set이 이미 주어져 있으므로, 사용할 수 있지만 다른 데이터 set은 없는 경우가 있다. 그런 경우, Cross-validation이나 training data를 split하여 Evaluation한다.

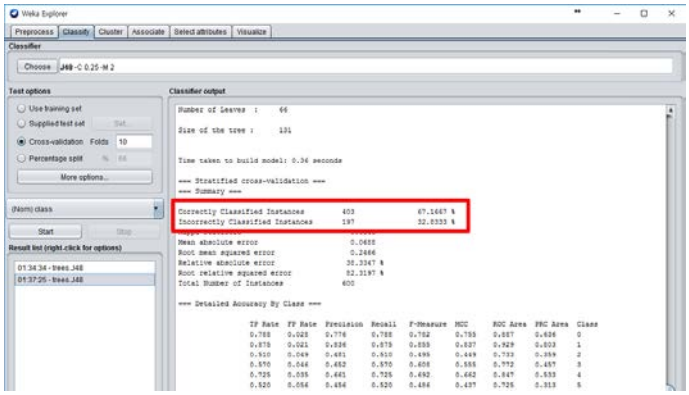
3.6.2. Cross-validation

-Cross-validation을 사용하지 않은 모델의 데이터 분류 정도



참고) training set과 test set의 데이터를 같게 놓고 모델을 테스트 하였다. 옳게 분류한 값이 많으나, 이것은 general 한 모델이 아님을 유추할 수 있다.

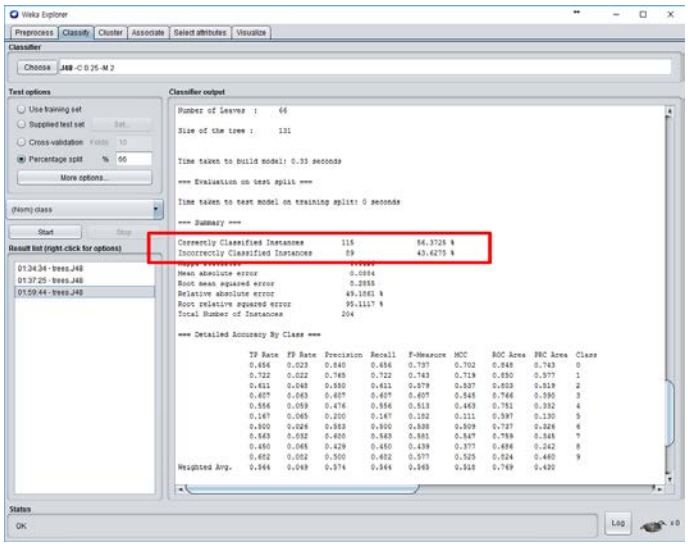
-Cross-validation을 사용한 모델의 데이터 분류 정도



참고) test option을 cross-validation으로 지정하였다.
10은 10 fold cross validation을 의미한다. 비교적 general 한 모델이 학습되었음을 알 수 있다.

3.6.3. Split

① Test option의 split 선택



참고) 66%로 나누어 66%는 training set, 나머지 44%는 test set으로 사용한다.

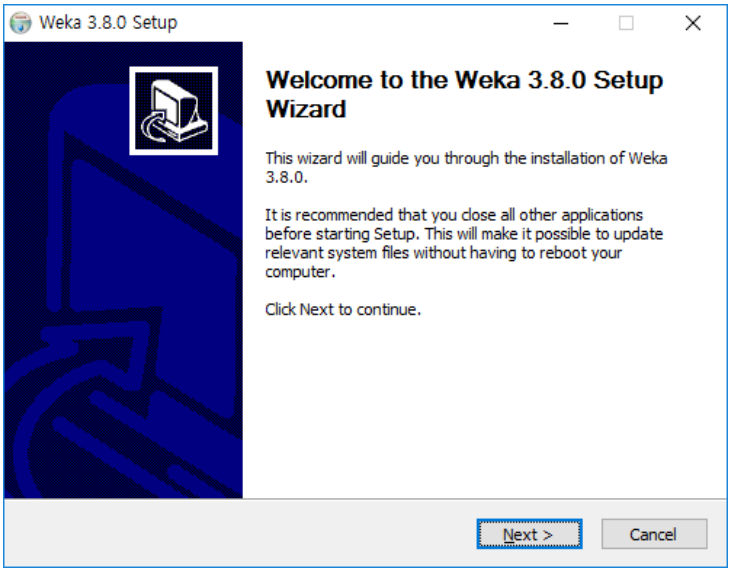
부록

4. 실습환경 설정

4.1. Weka

4.1.1. windows / mac 설치

http://www.cs.waikato.ac.nz/ml/weka/downloading.html 에 접속하여 설치파일(105.5MB)을 다운받아 설치한다. 아래와 같은 화면이 나오면 Next를 눌러 설치하면 된다.



4.1.2. ubuntu 설치

① 사이트 접속 후 설치 파일 다운로드
terminal 에서 설치 파일 위치로 이동 후

② $ java -jar weka.jar 입력

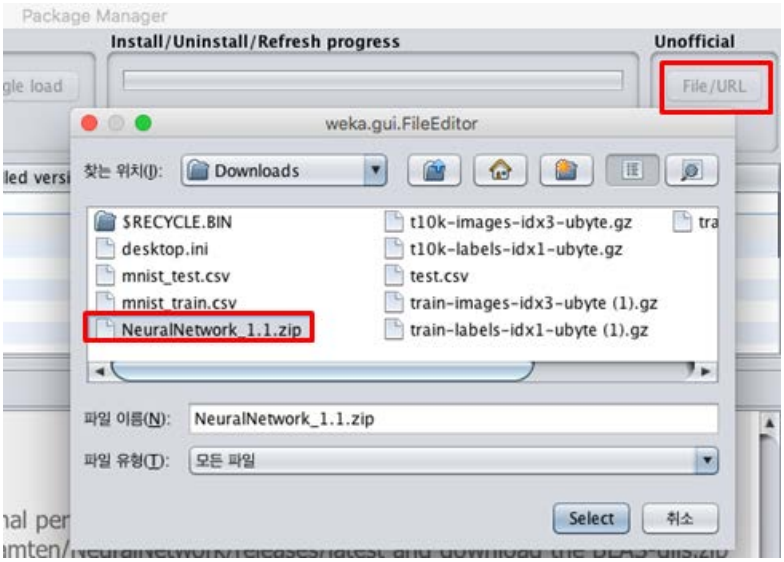
4.1.3. unofficial package 다운로드

③ 사이트 접속 후 필요한 패키지 다운로드
http://weka.wikispaces.com/Unofficial+packages+for+WEKA
cf) 공식 사이트에 없는 패키지 - Convolutional Neural Network package
https://github.com/amten/NeuralNetwork

④ Package Manager


Weka GUI Chooser
Program Visualization Tools Help
Package manager ^+U
ArffViewer ^+A
SqlViewer ^+S
Bayes net editor ^+N
KnowledgeFlow
Workbench
Simple CLI
Waikato Environment for Knowledge Analysis
Version 3.8.0
(c) 1999 - 2016
The University of Waikato
Hamilton, New Zealand

⑤ 패키지 적용


Package Manager
Install/Uninstall/Refresh progress
Unofficial
File/URL
weka.gui.FileEditor
찾는 위치(I):
Downloads
$RECYCLE.BIN
desktop.ini
mnist_test.csv
mnist_train.csv
NeuralNetwork_1.1.zip
t10k-images-idx3-ubyte.gz
t10k-labels-idx1-ubyte.gz
test.csv
train-images-idx3-ubyte (1).gz
train-labels-idx1-ubyte (1).gz
파일 이름(N):
NeuralNetwork_1.1.zip
파일 유형(T):
모든 파일
Select
취소