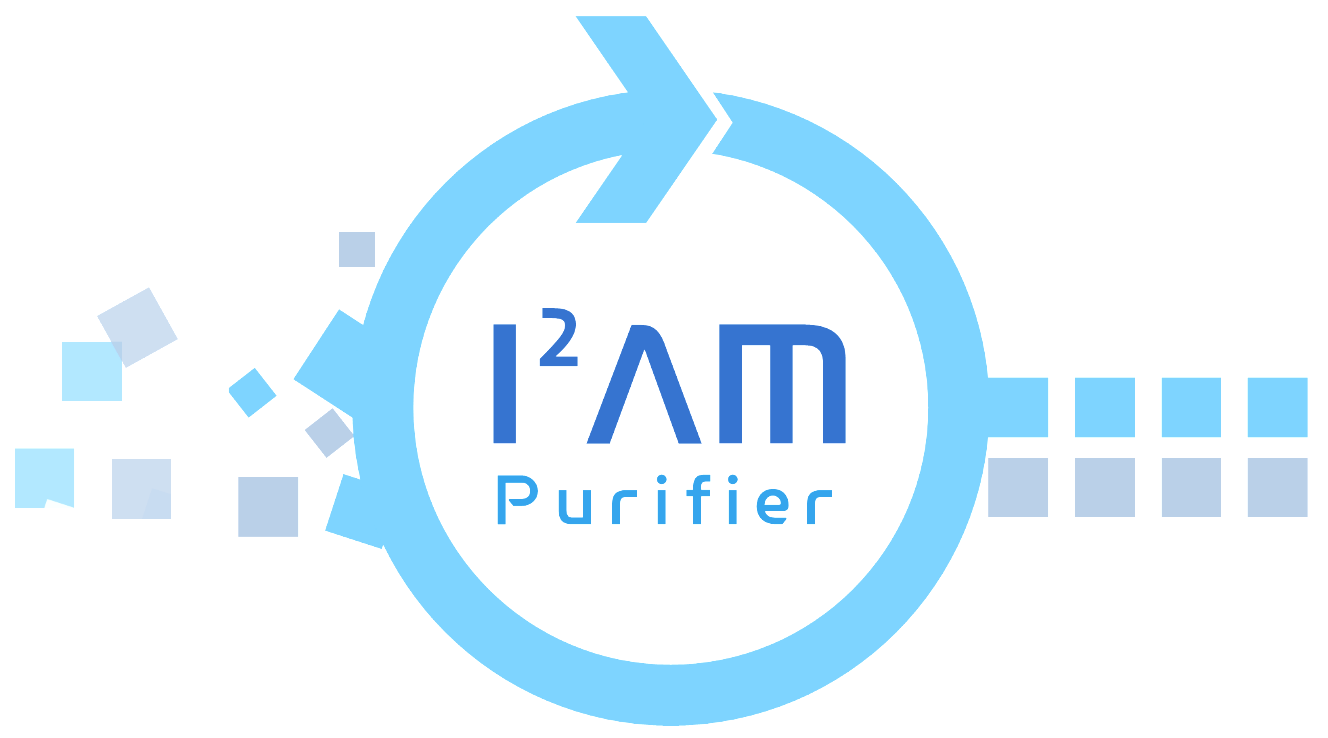
****

2016. 11.

연세대학교(원주캠퍼스), 강원대학교

지능형 시스템 구축을 위한

Machine Learning Open Library 분석서

문서 정보

|  |  |
| --- | --- |
| **버 전** | 0.4 |
| **작성일** | 2016-11-20 |
| **상 태** | 🞎 완료 🞎 진행 중 🗹 초안 |
| **작성자** | 김만수, 김대원, 이주원, 임효상 (연세대학교 원주캠퍼스) |
| **검토자** | 임효상 (연세대학교 원주캠퍼스) |
| **승인자** |  |

목 차

[Machine Learning 관련 1](file:///C:\Users\Daewon\Desktop\보고서_양식(ML%20LIbrary)(Draft)%20rev.2.docx#_Toc467166942)

[Open Library 분석서 1](file:///C:\Users\Daewon\Desktop\보고서_양식(ML%20LIbrary)(Draft)%20rev.2.docx#_Toc467166943)

[문서 정보 2](#_Toc467166944)

[1. Machine Learning Open Library 개요 1](#_Toc467166945)

[1.1. 소개 1](#_Toc467166946)

[1.2. 종류 1](#_Toc467166947)

[2. 개별 Machine Learning Open Library 분석 2](#_Toc467166948)

[2.1. Apache SAMOA 2](#_Toc467166949)

[2.1.1. 개요 2](#_Toc467166950)

[2.1.2. 제공되는 주요 알고리즘 3](#_Toc467166951)

[2.1.3. 설치방법 4](#_Toc467166952)

[2.1.4. 사용방법 및 사용 예 5](#_Toc467166953)

[2.2. Weka 5](#_Toc467166954)

[2.2.1. 개요 5](#_Toc467166955)

[2.2.2. 제공되는 주요 기능 및 알고리즘 6](#_Toc467166956)

[2.2.3. 설치방법 7](#_Toc467166957)

[2.2.4. 사용방법 및 응용 예 7](#_Toc467166958)

[2.3. TensorFlow 9](#_Toc467166959)

[2.3.1. 개요 9](#_Toc467166960)

[2.3.2. 특징 9](#_Toc467166961)

[2.3.3. 그래프 구조를 이용한 수치 연산의 표현 10](#_Toc467166962)

[2.3.4. 설치방법 11](#_Toc467166963)

[2.3.5. 사용방법 및 응용 예 11](#_Toc467166964)

[2.4. Apache Mahout 12](#_Toc467166965)

[2.4.1. 개요 12](#_Toc467166966)

[2.4.2. 데이터 포멧 13](#_Toc467166967)

[2.4.3. 제공되는 주요 알고리즘 13](#_Toc467166968)

[2.4.4. Apache Mahout 사용 예제 14](#_Toc467166969)

[2.5. Scikit-learn 15](#_Toc467166970)

[2.5.1. 개요 15](#_Toc467166971)

[2.5.2. 제공되는 주요 알고리즘 15](#_Toc467166972)

[2.5.3. 설치방법 16](#_Toc467166973)

[2.5.4. 사용방법 및 응용 예 16](#_Toc467166974)

[3. 결론 및 요약 18](#_Toc467166975)

[4. Reference 19](#_Toc467166976)

# Machine Learning Open Library 개요

## 소개

Machine Learning이란 인공지능의 한 분야로 컴퓨터가 학습할 수 있도록 하는 알고리즘과 기술을 개발하는 분야를 말한다.[[1]](#endnote-1) 최근 이와 같은 Machine Learning이 주목을 받으면서 Machine Learning Open Library도 빠르게 성장하고 있다. Machine Learning Open Library란 Machine Learning을 사용하고 싶은 사람 누구든지 효율적으로 사용 할 수 있게 다양한 알고리즘들이 정리되어 있는 라이브러리를 뜻한다. 다양한 종류의 Machine Learning Open Library가 존재하며 데이터 특성, 언어, 플랫폼 등과 같은 다양한 옵션에 맞추어서 본인에게 필요한 라이브러리를 선택하여 사용 할 수 있다.

## 종류

본 분석서에는 총 5개의 Machine Learning Open Library에 대해 소개한다. 각 라이브러리의 특징은 아래 표와 같다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 이름 | 개발주체 | 개발시기 | 제공하는 주요 기능, 알고리즘 | 지원 언어 | 특징 |
| Apache SAMOA | Apache Software Foundation | 2015.07 | 분류, 군집화, 회귀 분석,  빈발 항목 탐색 | Java | 스트림 기반  학습 가능 |
| Weka | The University of Waikato | 1999.04 | 분류, 데이터 전처리, 군집화, 회귀 분석, 특징 추출, 시각화 | Java | GUI, 광범위한  적용 범위 |
| TensorFlow | Google Brain Team | 2015.11 | 그래프 연산 | Python,  C++, etc | 높은 이식성 |
| Apache Mahout | Apache Software Foundation | 2015.04 | 군집화, 분류,  협업 필터링 추천시스템 | Java | 하둡 위에  적용됨 |
| Scikit-learn | David Cournapeau | 2007.06 | 분류, 군집화, 회귀 분석 | Python, Cython, C | 수학, 과학  분야에 유용함 |

# 개별 Machine Learning Open Library 분석

## Apache SAMOA

### 개요

Apache SAMOA는 Apache 소프트웨어 재단에서 개발중인 분산 스트림 데이터를 학습하는 자바 기반 분산 기계 학습(distributed streaming machine learning) 프레임워크이다. SAMOA는 스트림 프로세싱 알고리즘을 기술하기 위한 플랫폼 독립적인 구조를 가지고 있다. 이 구조를 기반으로 작성한 알고리즘은 Apache Storm, Apache Flink, Apache Samza 등과 같은 분산 스트림 처리 엔진(DSPEe)들에 그대로 적용 가능한 것이 특징이다.

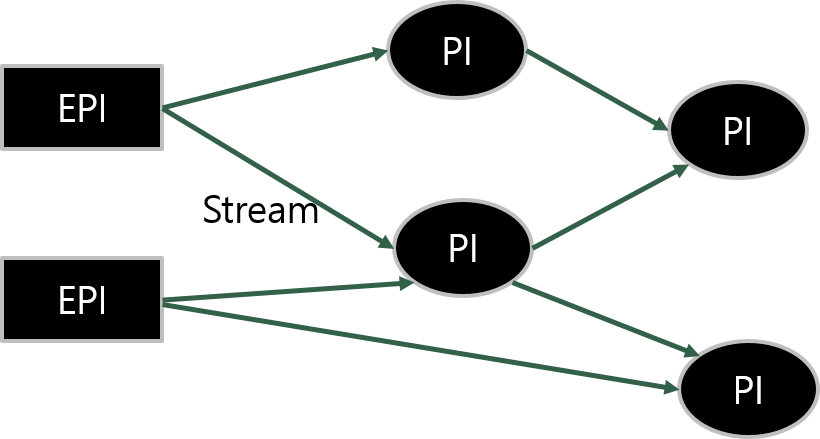


Figure 1. 분산 스트림 처리를 위한 SAMOA의 구조

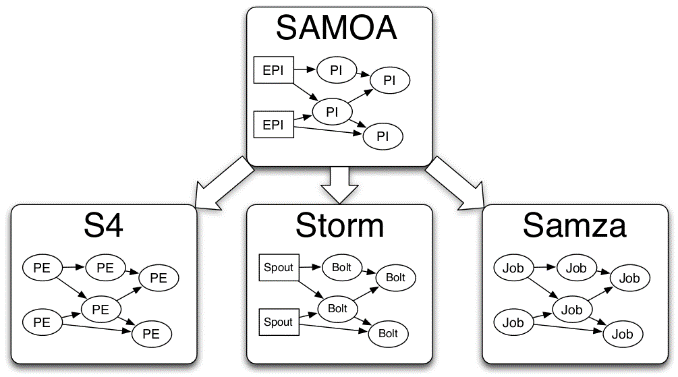


Figure 2. 분산 스트림 처리 엔진들에 그대로 적용 가능한 SAMOA 구조

### 제공되는 주요 알고리즘

* 분류

분류는 주어진 데이터들 각각을 여러 개의 부류 중 하나로 나누는 작업을 의미하며, Apache SAMOA는 데이터 스트림 환경에서의 분류 문제를 해결하기 위한 알고리즘으로 Hoeffding Tree(Very Fast Decision Tree)를 응용한 Vertical Hoeffding Tree Classifier를 제공한다.

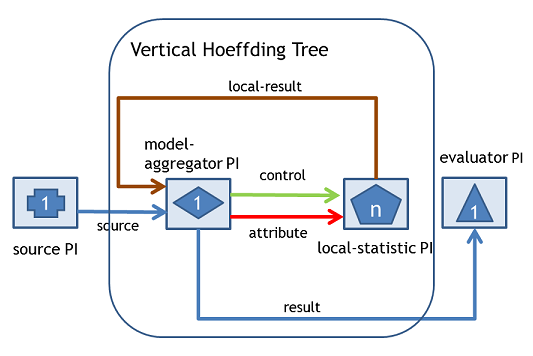


Figure 3. Vertical Hoeffding Tree (VHT) classifier

* 군집화

군집화는 주어진 데이터들을 분석하여 이 데이터들 사이에서 비슷한 데이터들을 그룹으로 묶는 과정을 말하며, Apache SAMOA는 Apache SAMOA Clustering Algorithm을 제공한다.

* 회귀 분석

회귀 분석은 주어진 데이터의 둘 이상의 속성의 상관관계를 파악하는 과정을 말한다. Apache SAMOA에서는 스트림 환경에서의 회귀 분석을 위한 Adaptive Model Rules Regressor를 응용한 Vertical Adaptive Model Rules Regressor를 제공한다.

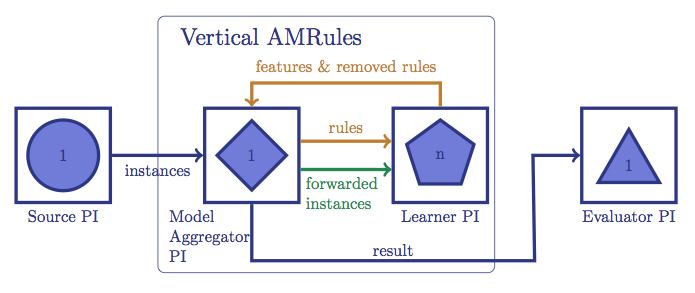


Figure 4. Vertical Adaptive Model Rules Regressor

* 빈발 항목 탐색

빈발 항목 탐색이란 주어진 데이터에서 빈발하는 아이템(항목) 세트를 찾아내는 과정을 의미하며, Apache SAMOA는 스트림 환경에서 빈발 항목 탐색을 위하여 PARMA 알고리즘을 응용한 Distributed Itemset Mining 알고리즘을 제공한다. 스트림 환경은 evolving nature를 포함하기 때문에, 과거의 빈발하는 항목이 현재에는 빈발하지 않을 수 있다. 이를 위해 SAMOA에서는 Time Biased Sampling 기법을 활용한다.

### 설치방법

* Linux 기준으로, JAVA 및 maven(자동 빌드 도구)의 설치는 생략한다.

1. git repository(http://git.apache.org/incubator-samoa.git) 의 SAMOA 소스코드를 로컬로 clone한다.
2. maven을 이용하여 사용할 플랫폼에 따라 SAMOA build를 수행한다. Build 시, incubator-samoa/target/SAMOA-\*.jar 형태로 target jar 파일이 생성된다.
3. SAMOA의 실행 파일은 설치 폴더 내 bin/samoa 이며 실행을 위해서는 인자로 타겟, 타겟 jar파일, task 및 옵션을 필요로 한다.

### 사용방법 및 사용 예

Apache SAMOA에 사용자 코드를 작성하여 사용하는 방법은 아래와 같다.

1. SAMOA 소스코드 폴더 하위에 자신의 java 소스코드를 입력한다.
2. Maven (Java build 관리 툴)을 이용하여 build를 수행하면 사용의 준비가 끝나게 된다.
3. SAMOA의 실행은 다음과 같은 형식의 명령어를 프롬프트에 입력하는 것으로 가능하다.

$(SAMOA 폴더에서)bin/samoa [타겟] [타겟 jar] “실행할 task 및 옵션“)

다음은 미국 산림청 자원정보 시스템에서 얻은 데이터를 분류하는 예제이다.

1. 미국 산림청의 샘플 데이터 다운로드 및 압축 해제를 수행한다.

$ wget "http://downloads.sourceforge.net/project/moa-datastream/Datasets/Classification /covtypeNorm.arff.zip"

$ unzip covtypeNorm.arff.zip

1. 로컬에서의 SAMOA 실행을 위해 아래와 같은 명령어를 입력한다.

$ bin/samoa local target/SAMOA-Local-0.3.0-SNAPSHOT.jar “PrequentialEvaluation -l classifiers.ensemble.Bagging -s (ArffFileStream -f covtypeNorm.arff) -f 100000"

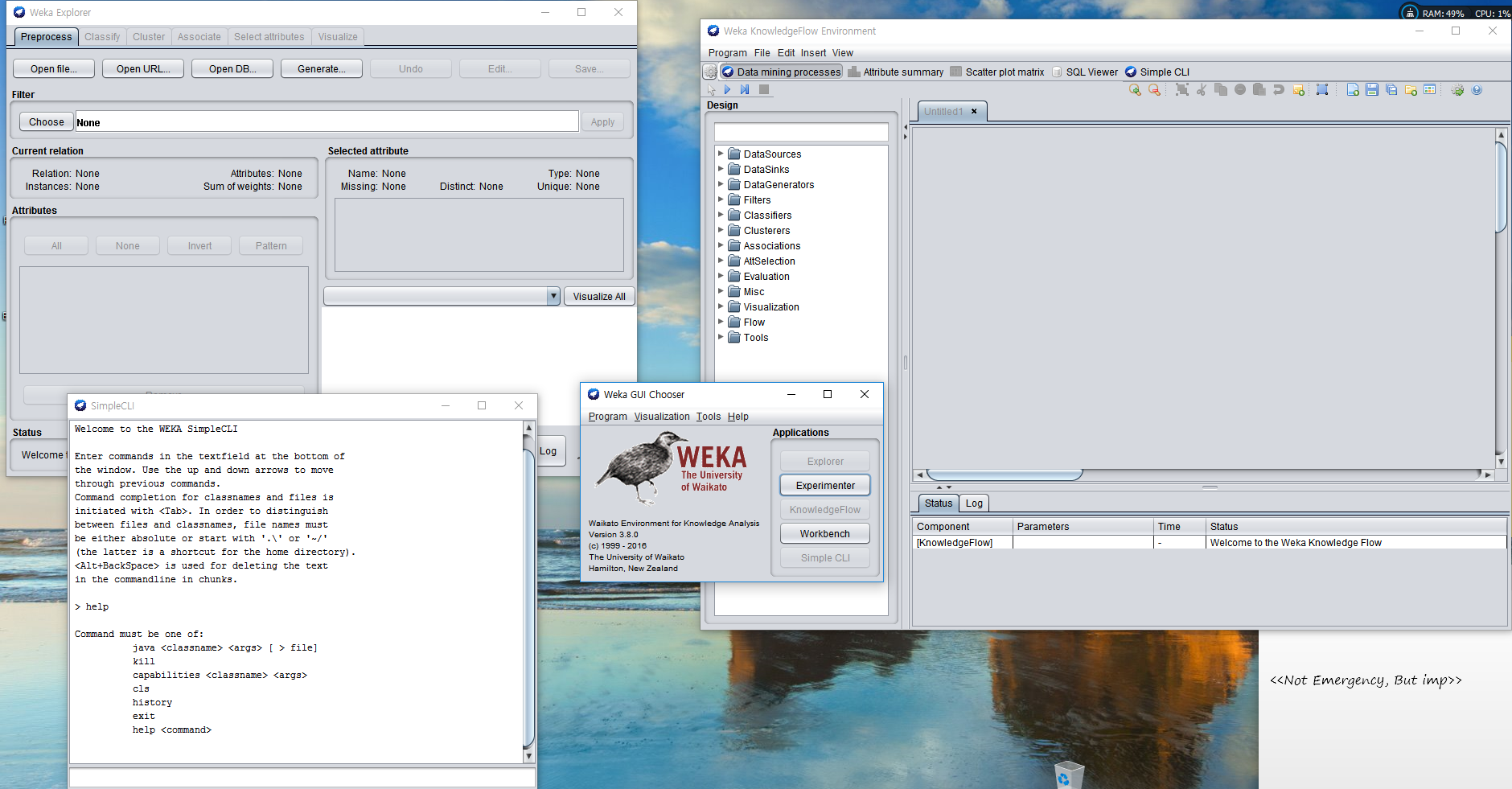
1. 분류 실행 결과는 아래 표와 같다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Evaluation instances | Classified instances | Classification correct(percent) | Kappa Statistic(percent) | Kappa Temporal Statistic(percent) |
| 100000.0 | 100000.0 | 70.938 | 36.07102307 | -210.3919684 |
| 200000.0 | 200000.0 | 74.6825 | 42.91969457 | -288.9315616 |
| 300000.0 | 300000.0 | 74.569 | 54.31275737 | -331.3019391 |
| 400000.0 | 400000.0 | 72.1895 | 51.85634712 | -407.1669554 |
| 500000.0 | 500000.0 | 71.4518 | 51.86477151 | -455.5421499 |

## Weka

### 개요

Weka는 뉴질랜드 Waikato 대학에서 데이터 마이닝 및 기계 학습 알고리즘 실험을 위해 개발한 프레임워크이다. Big Data를 기반으로 하는 범용 기계학습 알고리즘 및 툴 컬렉션을 제공하며, 기능의 추가가 용이하다. 다양한 데이터 소스(파일 포맷, 네트워크 파일, DBMS 등)를 지원하는 것으로 데이터 마이닝 및 기계학습 기술을 광범위하게 적용하는 것에 특화되어 있으며, GUI와 Data Visualization 기능을 통한 데이터 마이닝과 기계학습에 대한 높은 접근성을 제공한다.



**Figure 5. Weka**

### 제공되는 주요 기능 및 알고리즘

Weka는 GUI 및 커맨드 창에서의 작업을 지원하며, 툴 컬렉션을 통해 다음 기능 및 알고리즘들을 제공한다.

* 데이터 전처리

데이터 전처리란 데이터의 품질을 개선하거나, 데이터를 데이터 마이닝 분석에 적합한 형태로 변환하는 것을 말한다. 데이터 정제, 노이즈 제거 등의 작업이 이에 속한다.

* 군집화(Clustering)

군집화는 이질적인 개체들의 모집단으로부터 다수의 동질적인 하위 집단 혹은 군집으로 세분화하는 작업을 의미한다.[[2]](#endnote-2) Weka는 군집화 문제를 해결하기 위한 알고리즘으로 Hierarchical Clustering, k-means 알고리즘을 제공한다.

* 분류(Classification)

주어진 데이터가 여러 부류 중 어디에 속하는지 결정하는 문제를 분류 문제라고 한다. Weka는 분류 문제를 수행하기 위한 알고리즘으로 Naive Bayes, Bayesian Networks, RBF, k-NN, Logistic Regression, Decision Trees, Gaussian Process Classification 알고리즘을 제공한다.

* 회귀 분석(Regression)

회귀 분석이란 가장 적은 오류를 가지면서 데이터 전체를 대표할 수 있는 직선 혹은 곡선함수를 찾는 과정을 의미한다.[[3]](#endnote-3) Weka에서 회귀분석을 위해 제공하는 알고리즘으로는 Support Vector Regression, Gaussian Processes, Relevance Vector Machine 알고리즘이 있다..

* 특징 추출

특징 추출이란 마이닝의 결과로 생성할 모델에 적절한 속성 혹은 변수를 선택하는 과정을 의미한다.[[4]](#endnote-4) Weka에서는 특징 추출을 위해 Forward, Wrapper methods, Recursive Feature Selection 알고리즘을 제공한다.

* 데이터 시각화

데이터 시각화란 주어진 데이터를 그래프, 차트, 아이콘 등의 다양한 형태로 표현하는 것을 의미한다. 데이터 시각화 자체가 마이닝을 수행하는 것은 아니나, 데이터 시각화 결과는 데이터 마이닝 전에 데이터의 특성을 보여줌으로써 이후 마이닝 작업에 도움을 줄 수 있다.

### 설치방법

* Windows 기준으로, JAVA의 설치는 생략한다.

1. Weka 3.8(Latest stable version, 2016.5) 다운로드하고 실행 및 설치를 수행한다.

(<http://www.cs.waikato.ac.nz/ml/weka/downloading.html>)

1. 한글 데이터 사용을 위해 설치 경로의 RunWeka.ini 파일의 내용을 변경한다.

fileEncoding=UTF-8

1. 시작 – 모든 프로그램 – Weka 3.8.0 – Weka3.8.lnk를 통해 Weka GUI의 실행이 가능하다.

### 사용방법 및 응용 예

Weka 실행 시 Weka GUI Chooser가 나타나며, Weka GUI Chooser의 5가지 메뉴를 통해 데이터 마이닝의 수행이 가능하다.

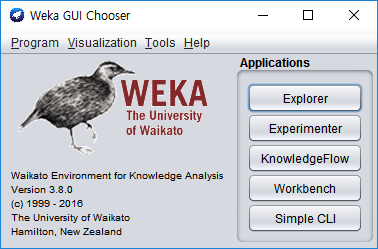


Figure 6. Weka GUI Chooser

* Explorer

특정 데이터에 대해 여러 옵션으로 마이닝 및 머신 러닝을 시험 해볼 수 있는 메뉴이다.

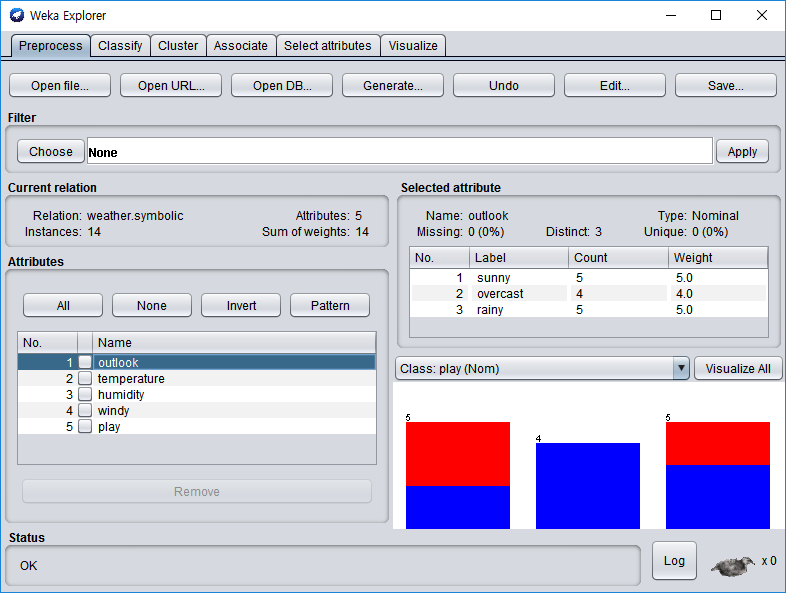


Figure 7. Weka Explorer를 이용한 데이터 전처리

* Experimenter

머신 러닝 알고리즘들의 성능 비교를 할 수 있는 메뉴이다.

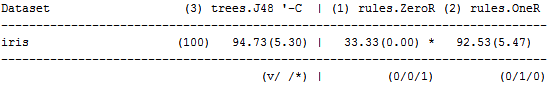


Figure 8. Weka Experimenter를 이용한 J48, ZeroR, OneR 알고리즘의 성능 비교

* KnowledgeFlow

기능은 Explorer와 동일하지만 드래그 앤 드롭 방식으로 마이닝 및 머신 러닝을 시험 가능한 메뉴이다.

* Workbench

3.8 버전에서 추가된 기능으로 위의 GUI들을 조합하여 사용하는 메뉴이다.

* Simple CLI

Command line 인터페이스를 여는 메뉴이다.

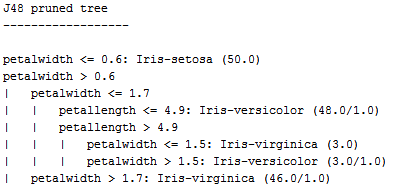


Figure 9. Weka를 이용하여 간단한 분류기를 생성한 모습  
(J48 알고리즘을 통해 붓꽃 데이터를 특징에 따라 분류)

## TensorFlow

### 개요

TensorFlow는 기계 지능의 연구를 위해 구글 브레인 팀에 의해 DistBelief의 후속으로 개발된 오픈 소스 소프트웨어 라이브러리이다.[[5]](#endnote-5) 모든 수치 연산을 데이터 플로우 그래프를 이용하여 표현하고 그래프 노드 간 데이터(Tensor) 교환을 통해 계산을 수행한다. C++ 기반이나 Python 언어 및 SWIG의 사용이 가능한 언어들을 추가적으로 지원한다.

### 특징

* 깊은 유연성

알고리즘을 그래프 형태로 표현하여, 그래프의 각 노드에 작은 연산을 정의하고 이 노드들을 서로 연결하여 알고리즘을 구현하는 방식의 프로그래밍을 지원한다.

* 연구와 (코드)생산의 연결

알고리즘을 코드로 변환하는 과정 대신, 알고리즘의 수식을 입력하는 것으로 알고리즘의 구현이 가능한 Symbolic 연산기능을 지원한다.

* 높은 이식성

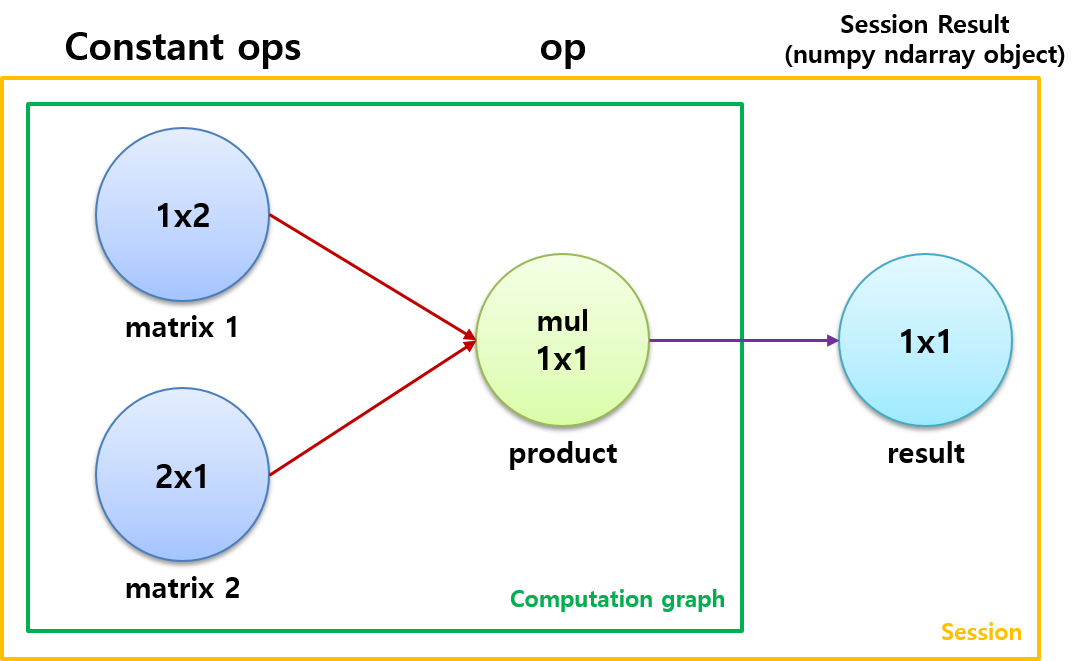
CPU에서 동작하도록 구현한 코드를 변경 없이 GPU에서 scale-up가능하며, 학습 완료된 모델을 모바일에서 활용하는 것이 가능하다.

* 성능 최대화

TensorFlow는 그래프 계산을 서로 다른 여러 계산 장치에서 함께 수행 가능하게 하며, 이를 통해 사용자의 하드웨어를 최대한 사용 가능하게 한다.

### 그래프 구조를 이용한 수치 연산의 표현

TensorFlow는 사용자가 필요로 하는 알고리즘을 그래프로 표현 가능하게 하여 사용자가 알고리즘 및 뉴런 네트워크를 쉽게 구현하도록 한다. 그래프로 표현된 연산을 연산 그래프라고 한다. 모든 종류의 데이터를 표현하기 위한 데이터 구조로 Tensor를 사용한다. Tensor는 n-차원 배열 혹은 리스트로 생각할 수 있다. 연산 그래프의 노드들은 ops(operations: 작업)라고 하며, 하나의 op는 0개 혹은 그 이상의 Tensor를 받아들여, 연산을 수행하고 0개 혹은 그 이상의 Tensor를 생성한다. 이와 같이 Tensor들이 연산 그래프의 op들을 지나가는 과정을 연산이라고 한다.



**Figure 10. 간단한 행렬 곱셉을 그래프로 표현한 예**

### 설치방법

* Linux 기준으로, JAVA 및 Python(3.5)의 설치는 생략한다.

1. Anaconda를 설치한다.
2. conda 환경변수를 지정하여 TensorFlow가 다른 Python 프로그램 및 라이브러리 버전들과 격리되도록 한다.
3. Python의 패키지 관리 시스템인 Pip을 이용하여 TensorFlow를 설치한다.

### 사용방법 및 응용 예

* Python 프롬프트에서 바로 사용
* >>> import tensorflow as tf
* Conda 환경 내에서 사용하는 경우
* TensorFlow 사용 🡪 $ source activate tensorflow
* TensorFlow 사용 종료 🡪 (tensorflow)$ source deactivate

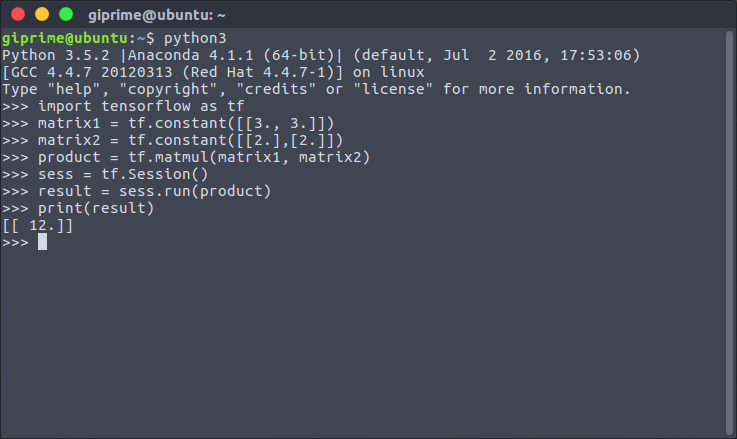


Figure 11. TensorFlow를 통해 간단한 행렬곱셉을 수행한 결과

## Apache Mahout

### 개요

Apache Mahout은 하둡 기반으로 분류, 클러스터링, 패턴 마이닝 과 같은 다양한 작업에 사용하기 위한 기계 학습 라이브러리 집합이다. 자바를 기반으로 하고 확장 가능한(scalable) 기계학습 라이브러리 구현을 목표로 한다. Apache Lucene의 하위프로젝트로 시작되었다가 오픈 소스기반의 협업필터링 프로젝트를 흡수하면서 Apache Mahout으로 발전되었다.[[6]](#endnote-6) 비슷한 특성을 가진 데이터들을 분류하고 정의하는 작업 및 협업 필터링 분야에 집중한다. 맵리듀스를 이용하는 아파치 하둡위에 적용되고 하둡의 독립노드(stand-alone)나 하둡이 없는 환경에서도 독립적으로 동작이

가능하다. 최근 맵리듀스 프레임워크에서 스파크 프레임워크로 코드 기반이 변경되고 있다. 아래의 표는 Mahout의 전반적인 소프트웨어 스택을 나타낸다.

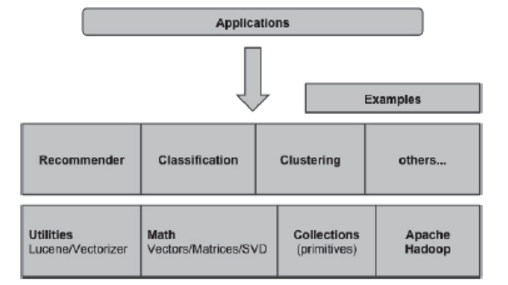


Figure 12. Physical View Mahout 소프트웨어 스택

### 데이터 포멧

Mahout은 텍스트 데이터나 숫자형 데이터 같은 원시 입력 데이터를 Mahout이 이해할 수 있는 형태로 변환이 필요하다. 이를 위해 Mahout은 벡터와 행렬 데이터 포맷을 사용한다. 이 포맷을 사용 할 때 데이터의 객체를 잘 표현하는 특성들을 선택하고, 객체의 특성을 컴퓨터가 이해할 수 있는 숫자형 또는 범주형 타입의 수학적 표현으로 바꿔주어야 한다. 입력 된 벡터와 행렬 데이터 포맷은 시퀀스파일(SequenceFile)형식으로 변환하여 사용한다. 시퀀스파일 형식은 하둡의 맵리듀스에서 입출력으로 사용되던 파일 포맷이다. 입력 데이터의 벡터 표현을 시퀀스 파일 형태로 변환(VectorWritable)해야만 Mahout 알고리즘을 수행할 수 있다.

### 제공되는 주요 알고리즘

* 군집화 (Clustering)

문서 또는 점(point)과 같은 많은 수의 객체를 유사성과 연관성이 높은 객체들로 구성하는 알고리즘이다. 이 과정에서 대용량 데이터로부터 흥미로운 정보를 가진 그룹을 발견할 수 있고, 데이터에 대한 직관을 얻을 수 있다. 실제로 서비스에서 숨겨진 사용자 그룹을 발견하거나 사이트 로그를 분석해 공통적인 사용 패턴을 발견하는 데 활용되고, 많은 문서를 이해하기 쉽도록 구조화하는데 사용된다. 군집화를 위해 K-Means, Fuzzy K-Means, Canopy, Meanshift, Hierarchical Clustering 등의 다양한 알고리즘이 제공된다.

* 협업 필터링 추천시스템

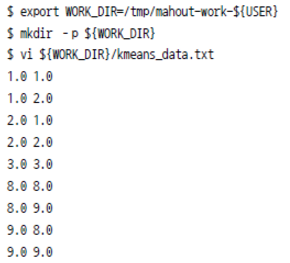
많은 사용자들로부터 얻은 기호정보(taste information)에 따라 사용자들의 관심사들을 자동적으로 예측해 주는 알고리즘이다. Memory, User, Item, Model 기반의 Collaborative Filtering 존재한다. 아마존, 넷플릭스, 페이스북, Last.fm 등 사용자와 사용자가 소비하는 아이템의 관계가 존재하는 많은 서비스에서 적극 활용되고 있다

* 분류 (Classification)

이미 특정 카테고리로 분류된 아이템들의 정보로부터 학습해 새로운 아이템을 정확한 카테고리로 분류하는 알고리즘이다. 오브젝트가 특정 카테고리에 종속되는지 또는 특정 속성을 포함하는지 결정 가능하다. 메일 서비스의 스팸 필터링 기법 또는 사진에서 사람의 얼굴을 구분하거나 이미지에서 문자를 구분하는 데 사용된다. 분류를 위해 (Complementary)Naive Bayes 분류기, Random Forests, Logistic Regression 등의 다양한 알고리즘을 지원한다.

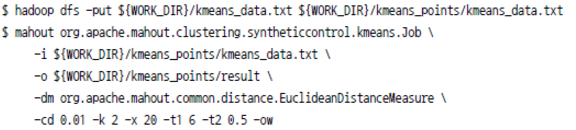
### Apache Mahout 사용 예제

Apache Mahout에서 군집화를 위한 K-Means Clustering 알고리즘을 사용하는 간단한 실습 예제이다. 하둡을 사용하지 않은 리눅스 환경에서 구현하였다. (x, y)좌표를 나타내는 9개의 샘플 데이터를 구성하였다. 샘플 데이터는 텍스트 파일 형태로 저장을 하였다.



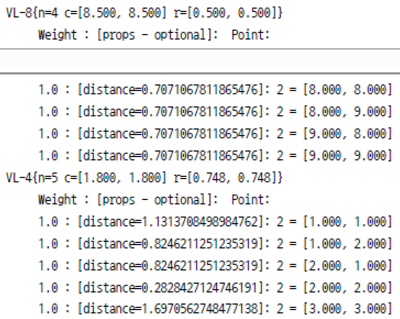
**Figure 13. 샘플 데이터 구성**

생성한 샘플데이터를 워킹 디렉토리로 삽입 한 후 알고리즘 및 옵션을 선택하였다. 알고리즘은 mahout에서 제공하는 K-Means 알고리즘을 선택하였다. 입력 데이터 및 출력 데이터를 설정하였고 거리 측정은 Euclidean 거리 측정 방법을 사용하였다.



**Figure 14. 알고리즘 수행**

알고리즘을 수행 결과 2개의 군집으로 나뉘었고 각 군집의 centroid 좌표를 확인 할 수 있다. 각 군집내에는 각각 4, 5개의 point를 포함하고 있다.



**Figure 15. 결과**

## Scikit-learn

### 개요

머신러닝을 위한 파이썬 오픈 소스 라이브러리이다. 2007년 David Cournapeau가 구글 섬머 코드 프로젝트로 시작하였다. NumPy와 SciPy를 기반으로 한다. SciPy를 보완하는 과학적 컴퓨팅 라이브러리들이 Scikit-xxx 형태로 이름 지어지고 machine learning 알고리즘을 제공하는 라이브러리이기 때문에 Scikit-learn이라고 불리게 되었다. 현재 30여개의 contributor들이 있으며 INRIA, Google, Python Software Foundation에서 유료 지원을 받고 있다. Supervised/Unsupervised learning 두 가지 모두를 지원한다. BSD license 형태의 오픈 소스이며 상업적으로 사용 가능하다.

### 제공되는 주요 알고리즘

* 군집화 (Clustering)

문서 또는 점(point)과 같은 많은 수의 객체를 유사성과 연관성이 높은 객체들로 구성하는 알고리즘이다. 이 과정에서 대용량 데이터로부터 흥미로운 정보를 가진 그룹을 발견할 수 있고, 데이터에 대한 직관을 얻을 수 있다. 실제로 서비스에서 숨겨진 사용자 그룹을 발견하거나 사이트 로그를 분석해 공통적인 사용 패턴을 발견하는 데 활용되고, 많은 문서를 이해하기 쉽도록 구조화하는데 사용된다. 군집화를 위해 K-Means, Fuzzy K-Means, Canopy, Meanshift, Hierarchical Clustering 등의 다양한 알고리즘이 제공된다.

* 협업 필터링 추천시스템

많은 사용자들로부터 얻은 기호정보(taste information)에 따라 사용자들의 관심사들을 자동적으로 예측해 주는 알고리즘이다. Memory, User, Item, Model 기반의 Collaborative Filtering 존재한다. 아마존, 넷플릭스, 페이스북, Last.fm 등 사용자와 사용자가 소비하는 아이템의 관계가 존재하는 많은 서비스에서 적극 활용되고 있다

* 분류 (Classification)

이미 특정 카테고리로 분류된 아이템들의 정보로부터 학습해 새로운 아이템을 정확한 카테고리로 분류하는 알고리즘이다. 오브젝트가 특정 카테고리에 종속되는지 또는 특정 속성을 포함하는지 결정 가능하다. 메일 서비스의 스팸 필터링 기법 또는 사진에서 사람의 얼굴을 구분하거나 이미지에서 문자를 구분하는 데 사용된다. 분류를 위해 (Complementary)Naive Bayes 분류기, Random Forests, Logistic Regression 등의 다양한 알고리즘을 지원한다.

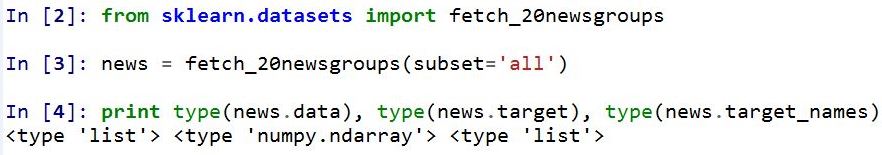
### 설치방법

SciPy기반으로 구현되었기 때문에 사용하기 위해서는 아래의 library가 필요하다.[[7]](#endnote-7)

* NumPy: 다차원 배열 등을 효율적으로 다루는 함수를 제공하는 라이브러리
* SciPy: 과학적 컴퓨팅을 위한 기본 라이브러리
* Matplotlib: 2D/3D 그래프를 그릴 수 있는 파이썬 라이브러리
* IPython: 파이썬 사용을 도와주는 커맨드 쉘
* Pandas: 데이터 로딩, 처리, 분석을 위한 파이썬 라이브러리

### 사용방법 및 응용 예

Scikit-learn 라이브러리를 이용하여 텍스트를 분류하는 예제이다.[[8]](#endnote-8) 기존에 존재하는 텍스트 문서에 대한 범주가 존재하고, 새롭게 텍스트 문서가 들어올 때 그 문서에 대한 범주를 예측하게 된다. 주로 스팸 필터에서 널리 사용된다. Classification의 대표적인 알고리즘인 Naive-Bayes를 사용하였고, Bayes’ rule을 적용하여 계산하였다. 훈련 데이터로 20개의 카테고리로 분류 된 뉴스그룹 문서를 사용하였다. 훈련 데이터의 개수는 총 18846개이다.



**Figure 16. 데이터 준비과정**

수집 한 훈련 데이터를 훈련 데이터와 테스트 데이터로 다시 나누는 전처리 과정을 진행한다. 전체 데이터 중 75%를 훈련데이터로 사용하고, 25%를 테스트 데이터로 설정한다.

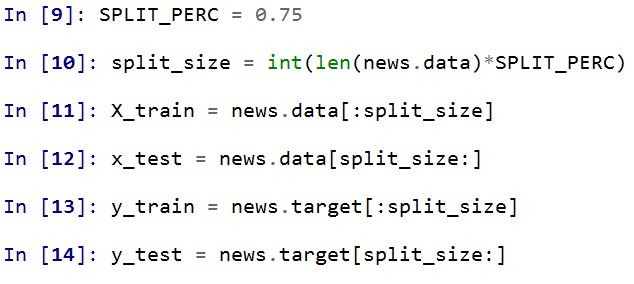


Figure 17. 데이터 전처리

나이브 베이즈 분류기 훈련을 위해 다항분포 클래스 및 특징을 추출하여 벡터라이저를 시키는 라이브러리들을 import하게 된다. Import한 세 개의 텍스트 벡터라이저를 각각 복합해 세 개의 다른 분류기를 생성하게 된다.

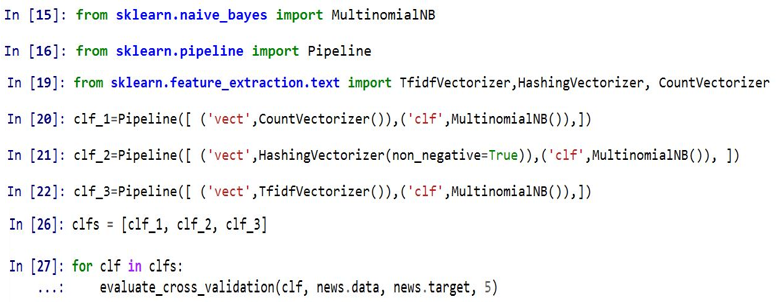


Figure 18. 나이브 베이즈 분류기 훈련

생성된 세개의 분류기 중 가장 좋은 성능을 내는 분류기를 사용하여 성능 평가를 진행할 수 있다. 성능 평가 시 precision, recall, f1-score, support 값 등을 확인 할 수 있다.

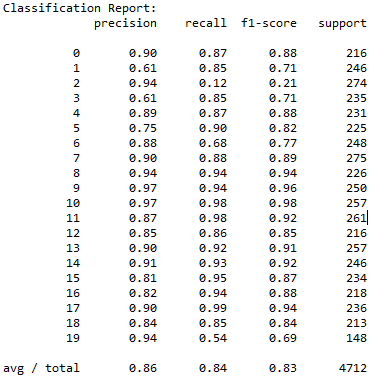


Figure 19. 성능평가

# 결론 및 요약

기계 학습의 구현을 위해 사용 가능한 여러 오픈 라이브러리에 대해 조사하였다. 라이브러리들은 각각 특징을 가지고 있어, 이후 시스템 개발 단계에 따라 선택적으로 활용 가능하다. GUI를 통해 기존 알고리즘을 범용 데이터에 활용 가능하게 하는 Weka를 이용하여, 시스템에 적용하고자 하는 기존의 알고리즘들을 빠르게 비교 및 테스트해볼 수 있다. Apache SAMOA를 통해 여러 분산 스트림 처리 환경에 대한 알고리즘 수행 코드를 한 번에 작성하여 분산 환경에서의 알고리즘 테스트가 가능하다. 구현하고자 하는 알고리즘을 개발 시스템 환경에 맞게 구현할 때, TensorFlow를 이용하면 그래프 형식으로 산술연산을 구현함으로써 복잡한 신경망의 구현을 직관적이고 효율적으로 할 수 있게 된다. Apache Mahout은 하둡과 스파크 기반 모두에서 유용한 라이브러리로써 배치 및 분산처리에 유용하다. Scikit-learn의 경우 코드 구현과 사용이 편리한 파이썬을 사용하기 때문에 간단한 예제 및 테스트에 유용하게 사용 될 수 있다.

# Reference

1. https://en.wikipedia.org/wiki/Machine\_learning [↑](#endnote-ref-1)
2. 지아웨이 한·미셸링 캠버·지안 페이, 데이터 마이닝 개념과 기법 Third Edition, 정사범·송용근 옮김, 에이콘출판주식회사(2015), p568 [↑](#endnote-ref-2)
3. SILBERSCHATZ, 데이터베이스 시스템 6th edition, 김형주 옮김, McGraw-Hill KOREA(2010), p833 [↑](#endnote-ref-3)
4. https://en.wikipedia.org/wiki/Feature\_selection [↑](#endnote-ref-4)
5. https://www.tensorflow.org/ [↑](#endnote-ref-5)
6. https://mahout.apache.org/ [↑](#endnote-ref-6)
7. http://scikit-learn.org/ [↑](#endnote-ref-7)
8. 라울 가레타, 『파이썬과 기계 학습』, 전철욱 옮김, 에이콘출판주식회사(2015), p56 [↑](#endnote-ref-8)