기상청

[STEP4 분석모형 구축]

머신러닝 모델 활용 분석

기상청

기상기후
빅데이터 분석 플랫폼

[분석교육] 머신러닝(Machine Learning)

[분석교육] 랜덤포레스트(Random Forest)

[분석교육] 그래디언트부스팅(Gradient Boosting)

[분석교육] 딥러닝(Deep Learning)

1. 랜덤포레스트 분석 수행 함수
2. 그래디언트부스팅 분석 수행 함수
3. 딥러닝 분석 수행 함수
4. 이 외 분석 수행 함수

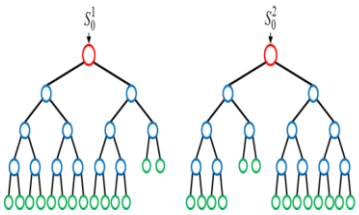
[분석 교육] 머신러닝(Machine Learning)(1/2)

머신러닝(Machine Learning) 또는 기계학습은 인공지능의 한 분야로, 컴퓨터가 학습할 수 있도록 하는 알고리즘과 기술을 개발하는 분야를 의미한다.

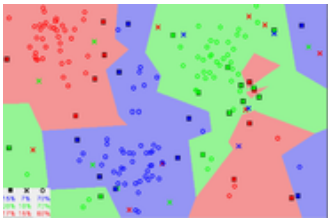
● 머신러닝의 개념

- 머신러닝(Machine Learning) 또는 기계학습은 컴퓨터가 학습할 수 있도록 하는 알고리즘과 기술을 개발하는 분야를 의미한다.
- 기계 학습과 데이터 마이닝은 종종 같은 방법을 사용하며 상당히 중첩되지만 다른 개념을 가진다.
  - 기계 학습은 훈련 데이터(Training Data)를 통해 학습된 알려진 속성을 기반으로 예측에 초점을 두고 있다.
  - 데이터 마이닝은 데이터의 미처 몰랐던 속성을 발견하는 것에 집중한다. 이는 데이터베이스의 지식 발견 부분의 분석 절차에 해당한다.

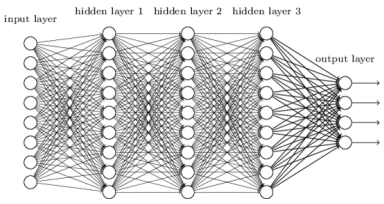
[다양한 머신러닝 알고리즘]



〈Random Forest〉



〈KNN Algorithm〉



〈Deep Neural Network :DNN〉

● 머신러닝의 활용 분야

- 사기 적발
- 설비고장 예측
- 텍스트 기반의 감성 분석
- 패턴 및 이미지 인식
- 이메일 스팸 필터링

[분석 교육] 머신러닝(Machine Learning) (2/2)

● 머신러닝 알고리즘의 종류

- 머신러닝 알고리즘은 크게 목표치를 가지고 훈련데이터로부터 하나의 함수를 유추하는 **지도 학습(Supervised Learning)**, 목표치 없이 데이터가 어떻게 구성되었는지를 알아내는 **자율 학습(Unsupervised Learning)**, 현재의 상태를 인식하여, 선택 가능한 행동들 중 보상을 최대화 하는 행동 혹은 행동 순서를 선택하는 **보강 학습(Reinforcement Learning)**으로 구분된다.

지도 학습	• Artificial neural network • Bayesian statistics • Gaussian process regression • Logistic Model Tree • Random Forests • Ensembles of classifiers • Ordinal classification • ANOVA • Linear classifiers • k-nearest neighbor • Decision trees • Bayesian networks • Hidden Markov models • …
자율 학습	• Expectation-maximization algorithm • Vector Quantization • Generative topographic map • Information bottleneck method • Artificial neural network • Association rule learning • Hierarchical clustering • Cluster analysis • Outlier Detection
보강 학습	• Temporal difference learning • Q-learning • Learning Automata • SARSA

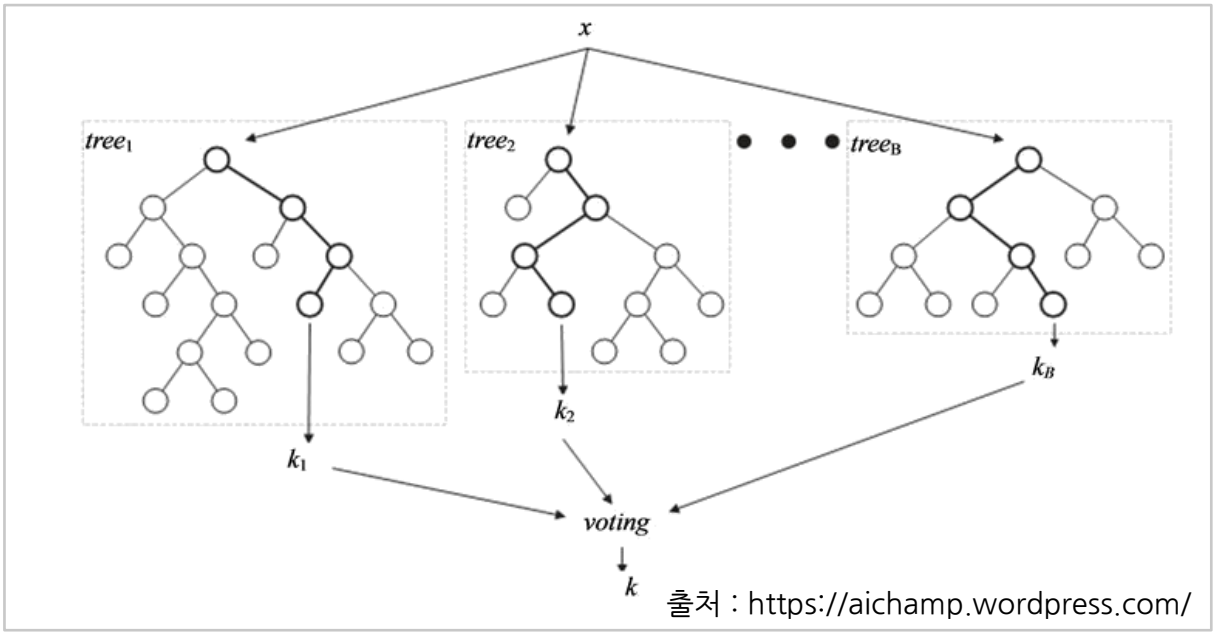
참고 자료 : 위키백과

[분석 교육] 랜덤포레스트(Random Forest)

랜덤포레스트(Random Forest)는 분류, 회귀 분석 등에 사용되는 앙상블 학습 방법의 일종으로, 훈련 과정에서 구성한 다수의 결정 트리로부터 분류 또는 평균 예측치를 출력하는 머신러닝 기법이다.

● 랜덤포레스트(Random Forest : RF)의 개념

- 랜덤포레스트(Random Forest : RF)는 주어진 데이터에 대해서 복원 샘플링을 통해 다수의 샘플 데이터를 생성하고 각 샘플 데이터를 모델링 한 후 결합하여 최종의 예측 모형을 산출하는 머신러닝 기법이다.



출처 : https://aichamp.wordpress.com/

● 랜덤포레스트의 특징

- 높은 예측력
- 변수 중요도 정보 제공
- 다수의 모형 결합을 통해 과대적합 방지
- 변수 수거 없이 수천 개의 독립 변수 적합 가능
- 종속 변수가 연속형일 때는 평균(Average), 범주형일 때는 다중 투표(Majority Vote)를 사용하는 것이 일반적

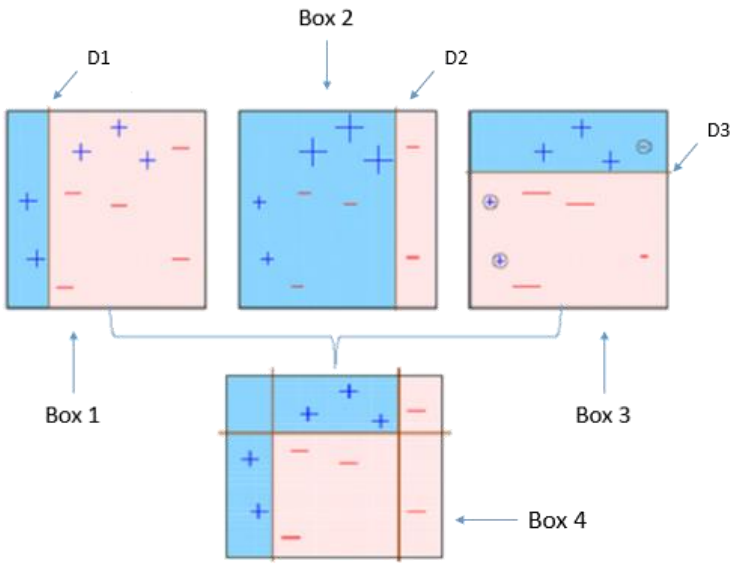
참고 자료 : 위키백과

[분석 교육] 그래디언트부스팅(Gradient Boosting)

그래디언트부스팅(Gradient Boosting)은 분류, 회귀 분석 등에 사용되는 앙상블 학습 방법의 일종으로, 훈련 과정에서 결정 트리를 연속적으로 학습하여 보다 강력한 학습기(learner)로 만드는 머신러닝 기법이다.

● 그래디언트부스팅(Gradient Boosting : GBM)의 개념

- 그래디언트부스팅(Gradient Boosting : GBM)은 의사 결정 트리의 연속적 학습을 통해 예측 모형을 생성하며, 연속되는 트리는 이전 트리의 예측 오류를 수정해 나가면서 모형의 예측력을 높이는 머신러닝 기법이다.



출처 : https://www.analyticsvidhya.com/

● 그래디언트부스팅의 특징

- 강한 예측력
- 변수 중요도 정보 제공
- 다수의 모형 결합을 통해 과대적합 방지
- 경사하강법(Gradient Descent)을 이용해 오류를 최소화하는 최적의 파라미터를 찾음
- 부스팅(Boosting) 알고리즘으로 모형의 정확도를 향상시킴

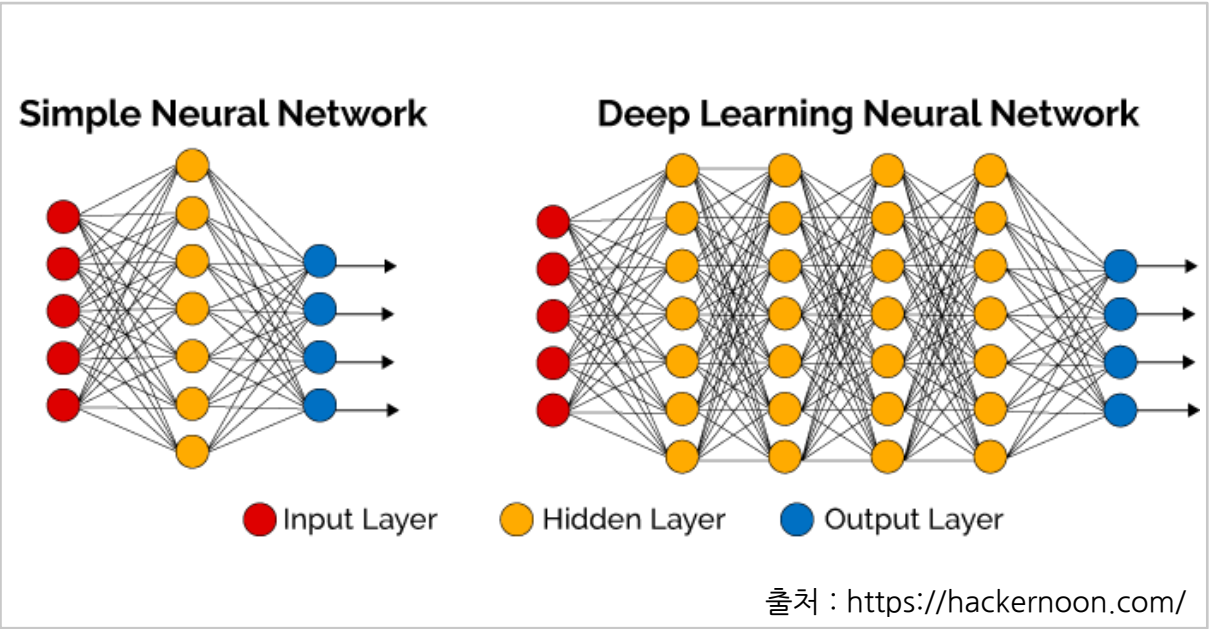
참고 자료 : 위키백과

[분석 교육] 딥러닝(Deep Learning)

딥러닝(Deep Learning)은 군집, 분류에 사용되는 머신러닝 기법의 일종으로, 여러 비선형 변환기법의 조합을 통해 높은 수준의 추상화를 시도하며, 큰 틀에서는 사람의 사고방식을 컴퓨터에게 가르치는 방법이다.

● 딥러닝(Deep Learning : DL)의 개념

- 딥러닝(Deep Learning : DL)은 인공 신경망을 층층이 쌓은 것으로, 입력층과 출력층 사이에 여러 개의 은닉층으로 이루어져 있으며, 비선형 변환 기법의 조합을 통해 예측모형을 산출하는 머신러닝 기법이다.



출처 : https://hackernoon.com/

● 딥러닝의 특징

- 복잡한 비선형 관계를 모델링하는데 탁월함
- 비지도 학습이 수행되는 여러 개의 층을 가짐
- 자동 특징 추출(feature extraction)로 데이터간의 유사성 파악이 쉬움
- 데이터의 잠재적인 구조를 파악
- 데이터의 양이 많으면 많을 수록 성능이 좋아짐

참고 자료 : 위키백과

1. 랜덤포레스트 분석 수행 함수

- **h2o::h2o.randomForest**
  - 랜덤포레스트 분석을 수행

▪ **Usage**

- training_frame : 학습 데이터 셋
- validation_frame : 검증 데이터 셋, 기본값은 NULL
- x : 독립변수
- y : 종속변수
- ntrees : 트리의 개수, 기본값은 50
- max_depth : 트리의 깊이, 기본값은 20
- sample_rate : 각 트리의 샘플 비율, 기본값은 0.632
- mtries : 트리를 나눌때 고려되는 변수의 수, 기본값은 -1
- seed : 랜덤 데이터 생성 시, 고정 시드

▪ **Examples**

```
Fit <- h2o.randomForest(y = y,
                        x = varList,
                        training_frame = data.hex,
                        ntrees = 512,
                        max_depth = 20,
                        sample_rate = 0.3,
                        mtries = -1,
                        seed=1)
```

```
> #MODEL RUNNING
> fit <- h2o.randomForest(y = y,
+                         x = varList,
+                         training_frame = data.hex,
+                         ntrees = 512,
+                         max_depth = 20,
+                         sample_rate = 0.3,
+                         mtries = -1,
+                         seed=1)
|=================================================================| 100%
```

2. 그래디언트부스팅 분석 수행 함수

- **h2o::h2o.gbm**

  - 그래디언트부스팅 분석을 수행

▪ **Usage**

- training_frame : 학습 데이터 셋
- validation_frame : 검증 데이터 셋, 기본값은 NULL
- x : 독립변수
- y : 종속변수
- ntrees : 트리의 개수, 기본값은 50
- max_depth : 트리의 깊이, 기본값은 5
- sample_rate : 각 트리의 샘플 비율, 기본값은 1
- seed : 랜덤 데이터 생성 시, 고정 시드

▪ **Examples**

```
Fit <- h2o.gbm(y = y,
               x = varList,
               training_frame = data.hex,
               ntrees = 50,
               max_depth = 5,
               sample_rate = 0.3,
               seed=1)
```

```
> #MODEL RUNNING
> Fit <- h2o.gbm(y = y,
+                x = varList,
+                training_frame = data.hex,
+                ntrees = 50,
+                max_depth = 5,
+                sample_rate = 0.3,
+                seed = 1)
|=================================================================| 100%
```

3. 딥러닝 분석 수행 함수

- **h2o::h2o.deeplearning**
  - 딥러닝 분석을 수행

▪ **Usage**

- training_frame : 학습 데이터 셋
- validation_frame : 검증 데이터 셋, 기본값은 NULL
- x : 독립변수
- y : 종속변수
- hidden : 은닉층 개수, 기본값은 [200, 200]
- input_dropout_ratio : 입력층의 drop out 비율(과적합 방지), 기본값은 0
- seed : 랜덤 데이터 생성 시, 고정 시드

▪ **Examples**

```
Fit <- h2o.deeplearning(y = y,
                        x = varList,
                        training_frame = data.hex,
                        hidden = c(200, 200),
                        input_dropout_ratio = 0,
                        seed=1)
```

```
> #MODEL RUNNING
> Fit <- h2o.deeplearning(y = y,
+                         x = varList,
+                         training_frame = data.hex,
+                         hidden = c(200, 200),
+                         input_dropout_ratio = 0,
+                         seed = 1)
 |=======================================================================| 100%
```

4. 이 외 분석 수행 함수(1/2)

- **h2o::h2o.grid**

  - 하이퍼 파라미터(Hyper-parameter) 리스트 조합별로 모형 구축

▪ **Usage**

- algorithm : 분석을 수행할 분석 알고리즘
- grid_id : 생성되는 grid의 id
- training_frame : 학습 데이터 셋
- x : 독립변수
- y : 종속변수
- hyper_params : 적용할 하이퍼 파라미터 리스트

▪ **Examples**

```
hyper_params <- list(sample_rate = c(0.3,0.4), max_depth=c(18,20,25,30),
                     ntrees = c(256,512), mtries=c(-1,1))

m <- h2o.grid(algorithm="randomForest",
              grid_id="rf_grid",
              training_frame = data.hex,
              x = varList,
              y = y,
              hyper_params=hyper_params)
```

```
> #하이퍼파라미터 조합만들기
> hyper_params <- list(
+   sample_rate = c(0.3,0.4),
+   max_depth=c(18,20,25,30),
+   ntrees=c(256,512),
+   mtries=c(-1,1)
+ )
> #조합 모형 돌리기
> m <- h2o.grid(algorithm="randomForest",
+               grid_id="rf_grid",
+               training_frame = data.hex,
+               x = varList,
+               y = y,
+               hyper_params=hyper_params)
|=================================================================| 100%
```

4. 이 외 분석 수행 함수(2/2)

- **h2o::h2o.getGrid**

  - h2o.grid로 구축한 모형의 하이퍼 파라미터와 요약된 결과를 조회

  ▪ **Usage**
  - grid_id : 조회할 그리드 객체
  - sort_by : 그리드의 모형들을 조회할 때 정렬 기준
  : logloss, residual_deviance, mse, auc, accuracy, precision, recall, f1 등
  - decreasing : 내림차순 정렬

  ▪ **Examples**
  h2o.getGrid(grid_id = "rf_grid", sort_by = "mse")

```
> #mse가 낮은 순으로 정렬하기
> sorted_grid <- h2o.getGrid(grid_id = "rf_grid", sort_by = "mse") #rf_grid
> sorted_grid.df <- as.data.frame(sorted_grid@ summary_table)
> head(sorted_grid.df)
  max_depth mtries ntrees sample_rate        model_ids                mse
1        20     -1    512         0.3  rf_grid_model_9   2.05149248076784
2        18     -1    512         0.3  rf_grid_model_8  2.051863098744141
3        18     -1    512         0.4 rf_grid_model_24  2.052102997863395
4        18     -1    256         0.4 rf_grid_model_16  2.052935567987779
5        18     -1    256         0.3  rf_grid_model_0 2.0549482994574446
```

- **h2o::h2o.getModel**

  - 구축한 모형을 호출함

  ▪ **Usage**
  - model_id : 조회할 구축 모형의 id

  ▪ **Examples**
  h2o.getModel(sorted_grid@model_ids[[1]])

```
> h2o.getModel(sorted_grid@model_ids[[1]])
Model Details:
==============

H2ORegressionModel: drf
Model ID:  rf_grid_model_8
Model Summary:
  number_of_trees number_of_internal_trees model_size_in_bytes min_depth max_depth
1             512                     512            59033566        18        18
  mean_depth min_leaves max_leaves mean_leaves
1   18.00000       5862      10844  9173.54100


H2ORegressionMetrics: drf
** Reported on training data. **
** Metrics reported on Out-Of-Bag training samples **

MSE:  2.050249
RMSE:  1.431869
MAE:  1.076704
RMSLE:  0.3754732
Mean Residual Deviance :  2.050249
```

본 문서의 내용은 기상청의 날씨마루(http://big.kma.go.kr) 내
R 프로그래밍 교육 자료입니다.