ISSN 2383-630X(Print) / ISSN 2383-6296(Online)
Journal of KIISE, Vol. 49, No. 11, pp. 1026-1031, 2022. 11
https://doi.org/10.5626/JOK.2022.49.11.1026

OANet: 데이터베이스 성능 예측을 위한 주의관심 메커니즘 기반 Ortho-At-tention Net

(OANet: Ortho-Attention Net Based on Attention Mechanism for Database Performance Prediction)

염 찬 호 † 이 지 은 † 박 상 현 ††
(Chanho Yeom) (Jieun Lee) (Sanghyun Park)

요 약 데이터베이스에는 수정할 수 있는 다양한 매개변수들이 있는데, 이를 Knob이라 한다. Knob들의 설정에 따라 데이터베이스의 성능이 상이하기 때문에 데이터베이스의 Knob을 튜닝 하는 것이 중요하다. 이 때 Knob 설정에 따른 데이터베이스 성능을 신뢰할 수 있고 신속하게 예측할 수 있는 모델이 필요하다. 하지만 Knob 설정이 같더라도 벤치마크를 수행하는 워크로드가 다른 경우 그 결과가 다를 수 있다. 따라서 본 논문에서는 주의관심 메커니즘을 기반으로 한 OANet을 제안함으로써 Knob뿐만 아니라 워크로드와 Knob 간의 연관성도 고려할 수 있도록 하였다. 그리고 제안한 모델의 성능을 확인하기 위해 기존에 사용하던 기계학습 기법들과 데이터베이스의 성능 예측 결과를 비교하였고 가장 높은 결과를 보임으로써 모델의 우수성을 검증하였다.

키워드: 데이터베이스, 딥러닝, 머신러닝, 주의관심 메커니즘, 소프트 직교 정규화

Abstract Various parameters in a database can be modified, which are called knobs. Since the performance of the database varies according to the settings of the knobs, it is important to tune the knobs of the database. And when tuning, a model that can reliably and quickly predict database performance according to the knob setting is needed. However, even when the knob setting is the same, the results may be different if the workload performing the benchmark is different. Therefore, in this paper, we propose an OANet using the attention mechanism so that the relationship between the knob and the workload can also be considered. Through experiments, the performance prediction results of the database were compared to various machine learning techniques, and the superiority of the model was confirmed by showing the highest score.

Keywords: database, deep learning, machine learning, attention mechanism, soft orthogonality regulariztion

· 이 논문은 2021 한국소프트웨어종합학술대회에서 'OANet: 데이터베이스 성능 예측을 위한 Ortho Attention Net'의 제목으로 발표된 논문을 확장한 것임
· 이 논문은 2022년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원(IITP-2017-0-00477, (SW 스타랩) IoT 환경을 위한 고성능 플래시 메모리 스토리지 기반 인메모리 분산 DBMS 연구개발)과 국토교통부의 스마트시티 혁신인재육성사업으로 지원을 받아 수행된 연구임

† 학생회원 : 연세대학교 컴퓨터과학과 학생
chanho0475@yonsei.ac.kr
jieun199624@yonsei.ac.kr
†† 종신회원 : 연세대학교 컴퓨터과학과 교수(Yonsei Univ.)
sanghyun@yonsei.ac.kr
(Corresponding author임)

논문접수 : 2022년 2월 28일
(Received 28 February 2022)
논문수정 : 2022년 5월 18일
(Revised 18 May 2022)
심사완료 : 2022년 6월 2일
(Accepted 2 June 2022)

정보과학회논문지 제49권 제11호(2022. 11)

1. 서 론

스마트 시티화가 진행됨에 따라 도시 곳곳에서는 다양한 장치와 센서를 통해 여러 데이터를 수집한다. 수집된 데이터는 도시 운영 및 시민에게 제공되는 서비스의 효율성을 향상시키기 위해 사용된다[1,2]. 이러한 빅데이터 환경에서는 방대한 양의 비정형 데이터가 생성되고 이를 효율적으로 처리할 수 있는 데이터베이스가 사용되고 있다[3-5].

데이터베이스는 수정할 수 있는 매개변수들(e.g. Rocks DB의 write-buffer-size, num-levels, compression- ratio 등)이 무수히 많고 이것들을 Knob이라 한다. 이 Knob들을 어떻게 설정하는지에 따라 데이터 베이스의 성능이 달라진다[6]. 이에 Knob을 튜닝하기 위한 여러 선행연구가 진행되어 오고 있다[6-9]. Knob 튜닝의 주요한 프레임워크는 최적화 알고리즘을 통해 Knob을 선정하고 알고리즘 내에서는 성능을 예측하는 다양한 기계학습 모델을 사용한다. [7]은 튜닝을 할 때 활용하는 성능 예측 모델로 Random Forest[10]를 사용하였다. [9]은 튜닝을 할 때 Xgboost[11]와 Random Forest를 활용하여 Knob을 선별하였다. 모델의 예측 정확도가 전체적인 튜닝 결과에 영향을 주기 때문에 본 논문에서는 높은 예측 정확도를 도출하는 모델 개발을 목표로 한다.

Knob들의 값이 같더라도 어떤 워크로드(Workload)에서 실행되는지에 따라 데이터 베이스의 성능이 다르다[6]. 이러한 Knob과 워크로드의 상호연관성을 반영하기 위해 본 논문은 주의관심 메커니즘[12]을 적용하고 벡터 데이터인 Knob을 매트릭스 데이터로 표현할 수 있는 방법론을 제안한다. 결과적으로 워크로드와 Knob과의 연관성을 학습하여 워크로드를 고려해 성능을 예측하는 모델을 제안한다.

본 논문은 모델의 우수성을 보여주기 위해, 실험을 통해 정량적, 정성적 평가를 진행했다. 실험을 위해 데이터베이스의 한 종류인 RocksDB에 대해 RocksDB의 성능 벤치마크로 사용되는 db_bench[13]를 통해 생성한 데이터셋을 사용했다. 또한, 다수의 기계학습 회귀 모델과의 회귀 정확도의 평가지표인 R-squared(R2), Pearson Correlation Coefficient(PCC), Concordance Index(CI), Mean Squared Error(MSE) 수치의 비교를 진행한 결과, 제안하는 모델이 우수한 성능을 보였다. 그리고 주의관심 메커니즘 가중치에 따른 그래프를 통해 Knob과의 상호연관성을 정성적으로 보였다.

2. Ortho Attention Net

본 논문에서는 Knob들과 워크로드 간의 연관성을 파악하여 결과 예측에 참고할 수 있도록 주의관심 메커니즘[12]과 Soft Orthogonality Regularization[14]을 적용한 OANet(Ortho Attention Net)을 제안한다.

2.1 모델 구조

OANet의 구조는 그림 1과 같다. Knob과 워크로드의 정보를 통해서 데이터베이스의 성능 지표(e.g. RocksDB의 데이터 처리 시간(TIME), 데이터 처리량(RATE), 쓰기증폭(WAF), 공간증폭(SA))를 예측하는 것을 목표로 하고 정의한 변수는 다음과 같다. x_i는 각 Knob에 대한 값이고 X는 이 값들을 모두 포함하는 벡터이다. $\hat{y}$는 모델을 통해 예측한 데이터베이스의 성능지표, 그리고 wk는 워크로드 정보를 가리키고 one-hot 벡터를 통해 표현한다. one-hot벡터의 차원수는 사용한 워크로드 종류 개수(m)이다.

$$X = \{x_1, x_2, \ldots, x_n\} \tag{1}$$

$$\hat{y} = OANet(X, wk) \tag{2}$$

본 논문에서 제안하는 모델은 벡터 데이터인 Knob에 대해 주의관심 메커니즘을 적용하기 위해서 매트릭스 데이터로 변환하는 과정을 거친다. σ는 sigmoid 활성화 함수이며 $W_L \in \mathbb{R}^{n \times g*k}$와 b_L는 각각 선형변환 과정에서의 가중치와 편향을 가리킨다.

$$h_X = \sigma(XW_L + b_L) \tag{3}$$

$$\tilde{h}_X = Reshape(h_X) \tag{4}$$

수식 (3)을 통해 선형 변환으로 차원을 늘리고, 수식 (4)와 같이 $h_X \in \mathbb{R}^{1 \times g*k}$를 $\tilde{h}_X \in \mathbb{R}^{g \times k}$로 $Reshape$한다. 이

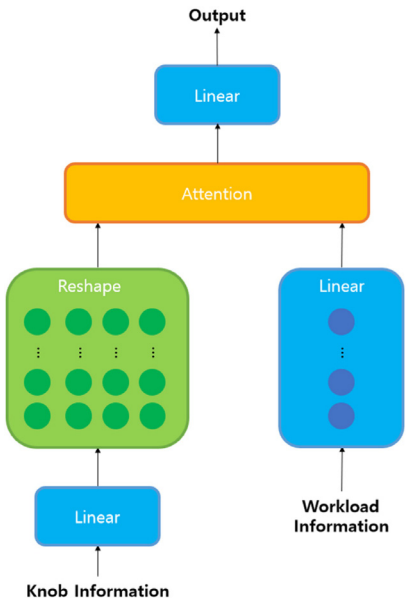

그림 1 모델 구조

Fig. 1 Model architecture

를 통해, Knob의 정보를 k차원을 갖는 g개의 벡터로 이루어진 매트릭스로 표현한다. 각 개의 벡터는 서로 독립적인 Knob의 특성을 반영해야 한다. wk는 주의관심 메커니즘에 활용하기 위해 수식(5)를 통해 선형 변환으로 $h_{wk} \in \mathbb{R}^{1\times k}$를 얻는다. $W_{wk} \in \mathbb{R}^{m\times k}$와 b_{wk}는 각각 선형변환 과정에서의 가중치와 편향을 가리킨다.

$$h_{wk=\sigma}(wkW_{wk}+b_{wk}) \quad (5)$$

Knob 정보와 워크로드 정보 사이의 연관성을 학습하기 위해 주의관심 메커니즘[12]을 적용한다. Q(query), K(key), V(value), d_k(key의 차원 크기)를 입력값으로 받는다.

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (6)$$

$$\alpha = \sigma(Attention(h_{wk}, \tilde{h}_X, \tilde{h}_X)) \quad (7)$$

$$\hat{y} = \alpha W_y + b_y \quad (8)$$

수식 (7)을 통해 워크로드 정보와 연관성이 높은 Knob의 특성을 학습하게 되고 특정 워크로드가 들어왔을 때 연관성이 높은 Knob에 대해 높은 가중치를 준다. 그리고 수식 (8)을 통해 데이터베이스의 성능지표를 예측한다. $W_y \in \mathbb{R}^{k\times 1}$와 b_y는 각각 선형변환 과정에서의 가중치와 편향을 가리킨다.

2.2 목적 함수

제안한 모델은 선형변환을 통해 주어진 Knob들로부터 의미있는 특성을 추출한다. 그리고 *Reshape*과정을 거쳐 그 특성들을 g개의 그룹으로 나눠 $\tilde{h}_X$를 구한다. 주의관심 메커니즘은 워크로드와 연관성이 높은 Knob 특성 그룹에 큰 가중치를 주고 이를 통해 성능 예측을 하므로 각 특성 그룹이 워크로드의 특징과 높은 상관관계를 가지도록 구분하는 것이 중요하다. 그를 위해 목적 함수에 *SOR* 정규항(Soft Orthogonality Regularization)을 추가한다. *SOR* 정규항은 인자로 들어온 매트릭스의 각 행벡터 간의 직교성을 나타내는 항으로 이를 목적함수에 포함하게 되면 $\tilde{h}_X$의 Knob 그룹들이 직교성을 가지도록 학습을 하게 되며 이는 다시 말해 특성 그룹이 워크로드의 특징과 높은 상관관계를 가지게 구분되도록 학습한다는 의미이다. 목적함수는 수식 (10)과 같이 계산된다.

$$SOR(\tilde{h}_X) = \lambda \left\| \tilde{h}_X \cdot \tilde{h}_X{}^T - I \right\|^2 \quad (9)$$

$$Loss = (1-\alpha)MSE(y, \hat{y}) + \alpha \cdot SOR(\tilde{h}_X) \quad (10)$$

3. 실험 및 결과

3.1 실험 환경

본 논문에서는 RocksDB에 대해 db_bench로 성능측정을 한 32만여개의 데이터 셋으로 실험을 진행하였다. 사용한 워크로드는 표 1에 나와있듯 Read-Write의 비율이 9:1, 1:1, 1:9 그리고 Update에 대해 value-size를 각기 다르게 한 16개이다. 괄호 안의 숫자는 value-size(KB)를 나타낸다. OANet 학습에 사용한 초매개변수(hyper parameter)는 g=32, k=16이다.

표 1 워크로드 종류
Table 1 Types of workload

RW 9:1 (1)	RW 9:1 (4)	RW 9:1 (16)	RW 9:1 (64)
RW 1:1 (1)	RW 1:1 (4)	RW 1:1 (16)	RW 1:1 (64)
RW 1:9 (1)	RW 1:9 (4)	RW 1:9 (16)	RW 1:9 (64)
Update (1)	Update (4)	Update (16)	Update (64)

3.2 실험 결과

본 논문에서는 제안하는 모델의 우수성을 입증하기 위해 다양한 실험을 진행하였다. 표 2의 실험을 통해 워크로드와 Knob간의 상호연관성을 파악하도록 하는 주의관심 메커니즘과 Knob의 특성을 잘 구분하게 학습하도록 하는 SOR 목적함수가 성능 향상을 이끌어냄을 검

표 2 주의관심 메커니즘과 SOR의 효용성 실험
Table 2 Effectiveness test for attention mechanisms and SOR

Model	Metric	R2 (↑)	PCC (↑)	CI (↑)	MSE (↓)
Single	**TIME**	0.9577	0.9787	0.9458	0.0425
	RATE	0.8767	0.9397	0.9034	0.1250
Attention(X)	**WAF**	0.9080	0.9533	0.9115	0.0939
SOR(X)	**SA**	0.9988	0.9995	0.9849	0.0012
OANet	**TIME**	0.9611	0.9806	0.9554	0.0391
	RATE	0.9242	0.9636	0.9175	0.0768
Attention(O)	**WAF**	0.9396	0.9702	0.9303	0.0616
SOR(X)	**SA**	0.9994	0.9998	0.9879	0.0006
OANet	**TIME**	**0.9803**	**0.9902**	**0.9603**	**0.0198**
	RATE	**0.9423**	**0.9708**	**0.9283**	**0.0585**
Attention(O)	**WAF**	**0.9514**	**0.9754**	**0.9346**	**0.0496**
SOR(O)	**SA**	**0.9999**	**1.0000**	**0.9928**	**0.0001**

증하였다. 주의관심 메커니즘과 SOR항 둘 다 적용하지 않은 단일 인공신경망인 Single 모델, SOR항은 적용하지 않고 주의관심 메커니즘만 적용한 2번째 모델 그리고 주의관심 메커니즘과 SOR항을 모두 적용한 OANet에 대해 데이터베이스의 성능지표 예측 결과를 확인하였고 그 결과 주의관심 메커니즘과 SOR를 적용함에 따라 성능이 좋아지는 것을 확인할 수 있다.

표 3은 OANet과 [7]에서 사용한 Random Forest, 기계학습에서 사용되는 모델인 SGD Regressor(linear regression with Stochastic Gradient Descent), SVR (Support Vector Machine), Gradient Boost, XGBoost와 성능을 비교 실험한 것이다. OANet이 다른 모델들과 비교하여 우수한 성능을 보인다.

표 3을 보면 전체적으로 모든 모델에서 SA의 예측 정확도가 다른 성능지표보다 높다. 그림 2는 SA를 예측할 때 워크로드별로 g개의 Knob 그룹에 대한 주의관심 가중치를 그래프로 나타낸 것이다. 특정 몇 개의 Knob 그룹에 대한 가중치가 높다. 이를 토대로 특정 몇 개의 Knob 그룹이 SA의 성능에 주된 영향을 미치기 때문에 상대적으로 예측하기 용이한 것으로 보인다.

또한 그림 2는 워크로드가 달라지면 Knob 특성 그룹에 대한 가중치가 달라지는 것을 확인할 수 있다. (a)와

표 3 OANet과 머신러닝 기법들과의 성능 비교

Table 3 Comparison of performance between OANet and machine learning techniques

Model	Metric	R2 (↑)	PCC (↑)	CI (↑)	MSE (↓)
SGD	TIME	0.7417	0.8613	0.8764	0.2596
	RATE	0.4752	0.6894	0.8230	0.5322
	WAF	0.6791	0.8246	0.8285	0.3274
	SA	0.8240	0.9079	0.9081	0.1760
SVR	TIME	0.9065	0.9516	0.9519	0.0961
	RATE	0.7499	0.8851	0.9114	0.2536
	WAF	0.8675	0.9328	0.9142	0.1352
	SA	0.9961	0.9981	0.9697	0.0039
Gradient Boost	TIME	0.8456	0.9260	0.9221	0.1552
	RATE	0.7559	0.8795	0.8700	0.2476
	WAF	0.7526	0.8799	0.8581	0.2525
	SA	0.9731	0.9908	0.9595	0.0269
XGBoost	TIME	0.9563	0.9788	0.9481	0.0439
	RATE	0.8908	0.9448	0.9151	0.1108
	WAF	0.8798	0.9411	0.9017	0.1226
	SA	0.9996	0.9998	0.9782	0.0004
Random Forest	TIME	0.9685	0.9842	0.9615	0.0316
	RATE	0.9122	0.9551	0.9341	0.0890
	WAF	0.9136	0.9561	0.9230	0.0882
	SA	**1.0000**	1.0000	**0.9994**	**0.0000**
OANet	TIME	**0.9803**	**0.9902**	**0.9603**	**0.0198**
	RATE	**0.9423**	**0.9708**	**0.9283**	**0.0585**
	WAF	**0.9514**	**0.9754**	**0.9346**	**0.0496**
	SA	0.9999	**1.0000**	0.9928	0.0001

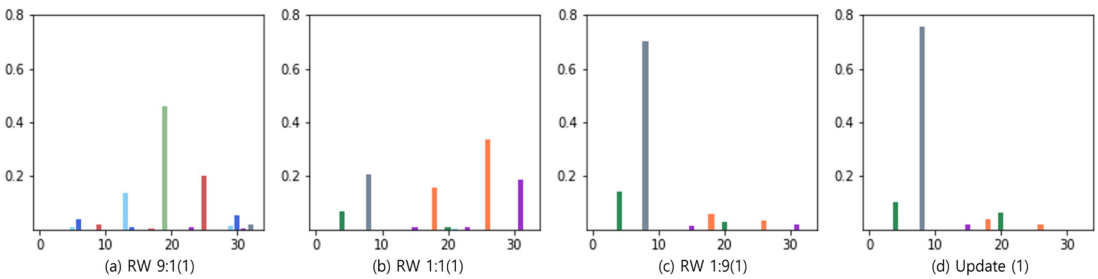

그림 2 SA예측 시 주의관심 가중치 그래프

Fig. 2 Attention weight graph when predicting SA

(b)의 경우 가중치를 높게 두는 Knob 특성 그룹이 다르다. [RW1:9 (1)]워크로드(c)와 [Update (1)] 워크로드(d)는 둘 다 공통적으로 쓰기의 비율이 높다. 그리고 해당 워크로드에 대한 주의관심 가중치를 보면 가중치를 높게 두는 그룹은 비슷하나 그 수치에서 차이가 있음을 확인할 수 있다. 이를 통해, Knob 특성 그룹이 잘 구분되도록 학습되었고, 워크로드와 Knob이 상호연관성이 있음을 보여준다.

또한 표 3의 성능 수치와 그림 2의 주의 관심 가중치 결과로부터 기존의 예측 모델들은 워크로드와 Knob 간의 관계를 고려하지 않고 주어진 Knob 설정에 대해서만 성능을 예측하나 OANet은 Knob과 워크로드 간의 관계를 파악하고 반영할 수 있기 때문에 다양한 워크로드에 대해 더 좋은 성능을 예측할 수 있음을 검증하였다.

4. 결 론

데이터베이스의 Knob 튜닝을 할 때 같은 Knob 수치에 대해서도 어떤 워크로드에서 수행되는지에 따라 결과가 상이하기 때문에 예측에 어려움이 있다는 문제점이 있다. 본 논문에서는 Knob과 워크로드간의 상호연관성을 파악할 수 있도록 주의관심 메커니즘 기반의 인공신경망을 제안하였다. 제안한 모델의 성능을 확인하기 위해 모델의 핵심 아이디어의 효율성을 검증하는 실험을 진행했고 기존에 사용되던 기계학습 모델들과의 성능을 비교하는 실험을 진행하여 더 우수한 성능을 보이는 것을 검증하였다. 본 논문에서는 RocksDB에 대해서만 실험 결과를 확인하였지만 다른 데이터베이스라 하더라도 본 논문에서 실험한 것과 같은 형식의 데이터셋을 만들 수 있기에 본 논문에서 제안하는 모델을 적용할 수 있다.

향후에는 본 논문에서 제안한 예측 모델을 통해 Knob 튜닝을 해서 데이터베이스의 성능을 향상시키는 연구를 진행해볼 예정이다.

References

[1] Mama Nsangou Mouchili, et al., "Smart City Data Analysis," *Proceeding of the First International Conference on Data Science, E-Learning and Information Systems*, Article No. 33, pp. 1-6, 2018.

[2] M. Dalla Cia et al., "Using Smart City Data in 5G Self-Organizing Networks," *IEEE Internet of Things Journal*, Vol. 5, No. 2, pp. 645-654, 2018.

[3] RocksDB. website. http://rocksdb.org/.

[4] Redis. https://redis.io

[5] MySQL. https://github.com/mysql

[6] Dana Van Aken et al., "Automatic Database Management System Tuning Through Large-scale Machine Learning," SIGMOD '17 : *Proceedings of the 2017 ACM International Conference on Management of Data*, pp. 1009-1024, 2017.

[7] Liu, Jiangyi, et al., "ATR: Auto-Tuning Configurations of Redis via Ensemble Learning," *2020 6th International Conference on Big Data and Information Analytics (BigDIA), IEEE*, pp. 104-112, 2020.

[8] Huijun Jin, Won Gi Choi, Jonghwan Choi, Hanseung Sung, Sanghyun Park, "A Study on the Analysis of RocksDB Parameters Based on Machine Learning to Improve Database Performance," *Proc. of the Korea Information Processing Society Conference 2020*, pp. 90-93, 2020. (in Korean)

[9] Juyeon Seo, Jieun Lee, Kyeonghun Kim, Jin Huijun, Sanghyun Park, "A Study on Redis Parameter Tuning Based on Non-linear Machine Learning," *Proc. of the KIISE Korea Computer Congress 2021*, pp. 69-71, 2021. (in Korean)

[10] Breiman, L, "Random forests," Machine learning, Vol. 45, No. 1, pp. 5-32, 2001.

[11] Chen, Tianqi, and Carlos Guestrin, "Xgboost: A scalable tree boosting system," *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016.

[12] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N. & Polosukhin, I. "Attention is all you need," *In Advances in neural information processing systems*, pp. 5998-6008, 2017.

[13] db_bench.github.com/google/leveldb/blob/main/benchmarks/db_bench.cc

[14] Bansal, Nitin, Xiaohan Chen, and Zhangyang Wang., "Can we gain more from orthogonality regularizations in training deep CNNs?," arXiv preprint arXiv:1810.09102, 2018.

이 지 은

2019년 인천대학교 컴퓨터공학부(학사)
2019~현재 연세대학교 컴퓨터과학과 석박사통합과정. 관심분야는 데이터베이스&기계 학습

염 찬 호

2021년 세종대학교 기계공학과(학사)
2021년~현재 연세대학교 컴퓨터과학과 석사과정. 관심분야는 데이터베이스&기계 학습

박 상 현

1989년 서울대학교 컴퓨터공학과 (학사). 1991년 서울대학교 대학원 컴퓨터공학과(공학석사). 2001년 UCLA 대학원 컴퓨터과학과(공학박사). 1991년~1996년 대우통신 연구원. 2001년~2002년 IBM T. J. Watson Research Center Post-Doctoral Fellow. 2002년~2003년 포항공과대학교 컴퓨터공학과 조교수. 2003년~2006년 연세대학교 컴퓨터과학과 조교수. 2006년~2011년 연세대학교 컴퓨터과학과 부교수. 2011년~현재 연세대학교 컴퓨터과학과 교수. 관심분야는 데이터베이스, 데이터 마이닝, 바이오인포매틱스, 빅데이터 마이닝 & 기계 학습