| 머신러닝 노트 | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|
| #1. 모델별 해당사항에 표시해보세요. | | | | | | | | |
|  | 딥러닝? | 지도? | 비지도? | 회귀? | 분류? | 군집화? | 변환? | 비선형? |
| Linear Regression |  |  |  |  |  |  |  |  |
| Ridge Regression |  |  |  |  |  |  |  |  |
| Lasso Regression |  |  |  |  |  |  |  |  |
| ElasticNet |  |  |  |  |  |  |  |  |
| Logistic Regression |  |  |  |  |  |  |  |  |
| Linear SVM |  |  |  |  |  |  |  |  |
| Kernel-SVM |  |  |  |  |  |  |  |  |
| Perceptron |  |  |  |  |  |  |  |  |
| MLP(FNN) |  |  |  |  |  |  |  |  |
| decision Tree |  |  |  |  |  |  |  |  |
| Random Forest |  |  |  |  |  |  |  |  |
| AdaBoost |  |  |  |  |  |  |  |  |
| GBM |  |  |  |  |  |  |  |  |
| Light GBM |  |  |  |  |  |  |  |  |
| Xgboost |  |  |  |  |  |  |  |  |
| Naive bayes Classifier |  |  |  |  |  |  |  |  |
| KNN |  |  |  |  |  |  |  |  |
| K-Means |  |  |  |  |  |  |  |  |
| Hierarchical-clustering |  |  |  |  |  |  |  |  |
| PCA |  |  |  |  |  |  |  |  |
| CNN |  |  |  |  |  |  |  |  |
| RNN |  |  |  |  |  |  |  |  |
| AE |  |  |  |  |  |  |  |  |
| GAN |  |  |  |  |  |  |  |  |

**< 직관에 도움이 되는 추상화 개념들>**

1. 머신러닝은 컴퓨터에게 데이터를 때려 넣음으로써 데이터 안의 패턴을 학습시키는 것이다. 그 학습의 목적은 새로운 데이터가 들어왔을 때, 그에 대한 예측을 가능케 하려는 것이다.
2. 머신러닝은 Cost 가 가장 낮을 때의 Weight 를 찾는 과정이다.
3. (2번->) 이게 최적화다.
4. Cost의 기울기(Gradient)는 Weight가 증가할 때 Cost가 어떻게 변하는지를 말해주는 수치다.
5. Weight의 수는 모델의 복잡도다. (모델의 복잡도는 Weight의 수에 비례한다.)
6. 규제는 모델의 복잡도를 제한하는 것이다.
7. R2\_score는 모델이 데이터의 분산도를 얼마나 설명하는지에 대한 수치다. (1 == 100%)
8. 정밀도를 잣대로 평가하면 모델은 positive라고 예측하기를 조심스러워하고, 재현율을 잣대로 평가하면 모델은 과감해진다.
9. 너 책 몇 epoch 돌렸어?
10. 너 한번 자리에 앉으면 학습하는 batch\_size가 어떻게 돼?
11. 디시젼 트리는 선택지 후보와 정답 후보가 있는 상황에서 푸는 스무(Max-depth)고개다.
12. 혼자보단 앙상블이 낫다. 집단지성의 힘, 공동체의 힘
13. K-means는 편가르기와 중심 재설정의 반복이다.
14. PCA는 어느쪽으로 봤을 때 데이터가 가장 많이 흩어져 보이는지를 찾겠다는 것이다.
15. 딥러닝은 결국 비선형 문제를 선형 문제로 바꾼다음, 선형 모델(단층 퍼셉트론)으로 풀겠다는 것이다.
16. CNN은 데이터의 공간적인 정보를 잘 반영하도록 설계한 구조이고, RNN은 시간적인 정보를 잘 반영하도록 설계한 구조다.
17. 딥딥익선. (딥러닝 연구자들의 성향 : layer 깊게 쌓고 싶어)

**< 향후 추천드리는 학습 방법 >**

1. 복습 및 개념 학습을 하고 싶다면, **모델 구현 코드의 파라미터를 수정**해가면서 결과를 비교해보시는 것이 가장 빠릅니다.
2. 모델에 대한 이해로는 역시 **Weight 개수 세보기**가 최곱니다. 복습으로도 아주 효과적입니다.
3. 모델에 대한 더 깊은 이해를 원할 때는 **raw 코드 뜯어보기**를 시전해보세요. 나름 재밌습니다. 코드실력도 팡팡
4. 근데, 사실 이보다는 **팀프로젝트**를 통해 **자유프로젝트, 공모전, 데이콘, 캐글**을 하시길 추천드립니다. 포트폴리오도 함께 쌓이는 게 좋으니까요.
5. **파이썬 프로그래밍 연습**은 그냥 매일 꾸준히 하세요. **암기 + 논리적 사고력**