|  |  |
| --- | --- |
| 교육 제목 | **결정트리** |
| 교육 일시 | 2021.10.20 |
| 교육 장소 | C6 |
| **교육 내용** | |
| 오전 | **\*결정트리:**  -예/ 아니오레 대한 질문을 이어나가면서 정갑을 찾아 학습하는 알고리즘  = 비교적 예측 과정을 이해하기 쉽고 성능도 뛰어남  -데이터 불러오기    -Data와 target 설정    -Split    -----------------------------------------------------------  -DecisionTreeClassifier() 클래스 사용하여 결정 트리 모델 훈련    -from sklearn.tree import plot\_tree  **\*가지 치기 = 푸르닝 /오버 피팅 제한 하는 방법**  - 트리의 깊이를 지정 = depth()    Plot tree = 트리 그림 그려 줌 , filled= 그림 색채워줌    특성 중요도는 결정 트리 모델의 feature\_importances\_ 속성에 저장    **\*cross\_validation = 교차검증**  - 안정적인 검증 점수를 얻고 훈련에 더 많은 데이터를 사용할 수 있음  - 훈련세트를 여러 폴드로 나눈 다음 한 폴드가 검증 세트의 역할을 하고 나머지 폴드에서는 모델을 훈련  - 교차검증은 이런 식으로 모든 폴드에 대해 검증 점수를 얻어 평균하는 방법    -교차 검증 함수= from sklearn.model\_selection import cross\_validate  import StratifiedKFold    -Mean()- 평균값 |
| 오후 | \*Grid search  - hyper parameter 탐색을 자동화 해주는 도구 /  parameter는 컴퓨터가 정해 줌  -탐색할 매개변수를 나열하면 교차검증을 수행하여 좋은 검증 점수의 매개변수 조합을 선택, 마지막으로 이 매개변수 조합으로 최종 모델을 훈련  from sklearn.model\_selection import GridSearchCV        \*random forest  - 결정트리 기반의 앙상블 학습 방법, 부트 스트랩 샘플을 사용하고 랜덤하게 일부 특성을 선택하여 트리를 만듬  - 트리를 작게 여러개  - 그래이디언트 부스팅: 결정 트리를 연속적으로 추가하여 손실함수를 최소화 하는 방법 , 훈련속도가 조금 느리지만 더 좋은 성능  -from sklearn.ensemble import RandomForestClassifier  Rf = 지정  -from sklearn.ensemble import GradientBoostingClassifier  Gf= 지정해서 똑같이 실행하면 된다 (실습x)        \*KMeans  - 랜덤 포인트가 어디로 찍히느냐에 따라서 결과가 달라짐  - 랜덤seed를 100 번해서 가장 많이 나온 쪽 - local minimum  - knn 이랑 매트리스 똑같음  - correlation = 패턴을 읽음  - euclidean = 거리      Install      사과 데이터 준비  -imshow() = 넘파이 배열로 저장된 이미지 그리기  흑백이미지 이므로 cmap매개변수로 gray 오 지정 , \_r로 반전시켜 줌    -subplot()함수사용 여러 개의 그래프를 배열처럼 쌓을 수 있도록 도와 줌 |