CH1

numpy

- 데이터 타입 : 숫자 문자열 불
  ndarray: 같은 데이터 타입만 가능, astype으로 타입 변환
- reshape : 차원과 크기 변경
- [ ] 이용해서 인덱싱과 슬라이싱
- 불리안 인덱싱 : [ ] 안에 조건문 추가하여
- sort : 정렬
- argsort : 정렬 행렬의 인덱스 반환
- dot : 내적 , transpose : 전치 행렬

pandas

- 넘파이보다 데이터 핸들링 편함
- 파일 포맷 변환 API : read_csv(콤마) read_table(탭)
- info()
  → 총 데이터 건수, 데이터 타입, null 건수
- describe()
  → only 숫자형 데이터의 사분위수, 평균, 표준편차, 최대, 최소
- value_counts() → 데이터 값 분포 확인
- Series: 인덱스와 단 하나의 칼럼(값)으로 구성된 데이터 세트
- DataFrame : 행과 열을 가지는 2차원 데이터
  → ndarray, 리스트, 딕셔너리로 상호 변환 가능
- df 데이터 생성/수정 : []
- df 데이터 삭제 : drop
  - axis 0:행, 1:열

- inplace=True 자신의 df에서도 데이터 삭제

- Index 객체
  - 만들어진 df/series index 객체는 변경 불가
  - reset_index → 새롭게 인덱스(기존 인덱스는 index 변수)

데이터 셀렉션 및 필터링

- [ ] → 칼럼 지정 연산자
- 명칭 기반 인덱싱 : 칼럼 명칭 기반으로 위치 지정
- 위치 기반 인덱싱의 구분 : 좌표 기반의 행열 위치를 통해
- iloc[] : 위치 기반 인덱싱, 행과 열에 인덱싱 값 입력 ex) iloc[0, 1]
- loc[]: 명칭 기반 인덱싱, 행에는 index값, 열에는 칼럼 명 입력 ex)
  data_df_reset.loc[1, 'Name']
  ** 행에 문자도 가능
- xi[] →둘 다 가능한데 지원 종료된 듯
- 명칭 기반 슬라이싱 : 시작점~종료점 포함
- 불린 인덱싱 → 여러 조건문을 통해 쉽게 추출

정렬

- sort_values (by=[,,,])
- 내림차순→ ascending=False
-

Aggregation 함수

- min max count sum mean

GroupBy

- 대상 칼럼에 따라
- Aggregation 추가 가능 ex) titanic_df.groupby('Pclass')['Age'].agg([max, min])

NA 처리

- is.na
- fill.na → NA 다른 값으로 대체

Apply lambda

- 함수를 좀 더 간단하게 구현
- lamda 이용해서 여러 값 입력 인자로 사용 시 map 사용
- else if는 지원x
- 함수를 받는 것도 가능 ex) apply(lambda x : get_category(x))

CH2

사이킷런

- 대표적인 파이썬 기반 머신러닝 개발 라이브러리

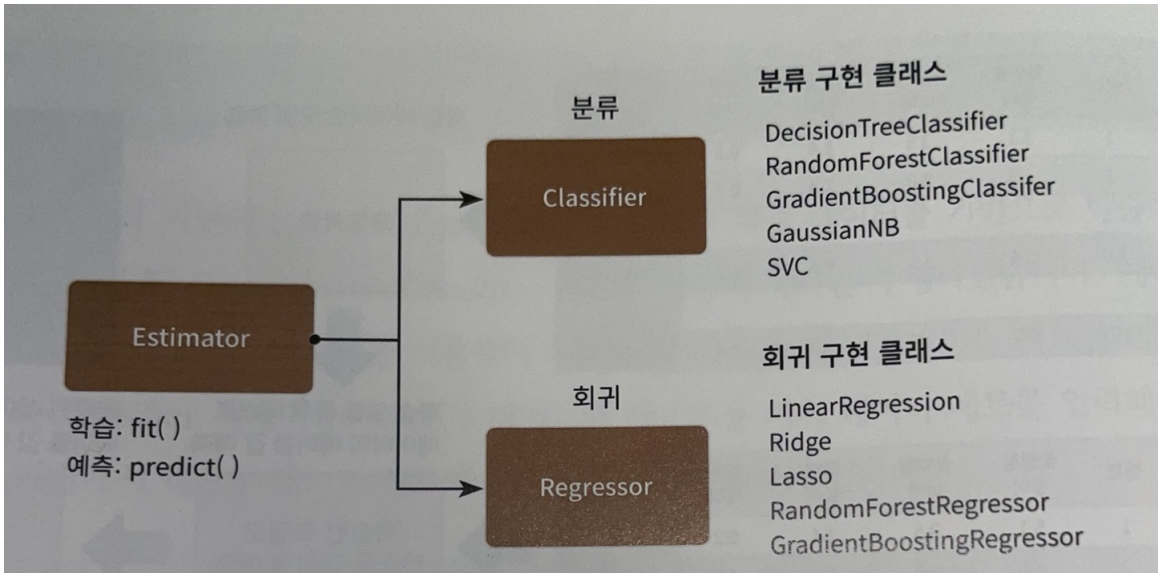
붓꽃 품종 예측

- 분류 → 대표적 지도학습 : 정답 주어진 문제 학습 후 새로운 문제 정답 예측
- iris 데이터 이용
- train_test_split : 학습용 데이터와 테스트 데이터 분리
- DecisionTreeClassifier 이용하여 예측
- accuracy_score : 정확도 측정

분류 예측 과정

1. 데이터 세트 분리
2. 모델 학습
3. 예측 수행
4. 평가

- fit : 모델 학습
- predict : 예측
- Estimator 클래스 → 주로 Classifier , Regression
- cross_val_score / GridSearchCV



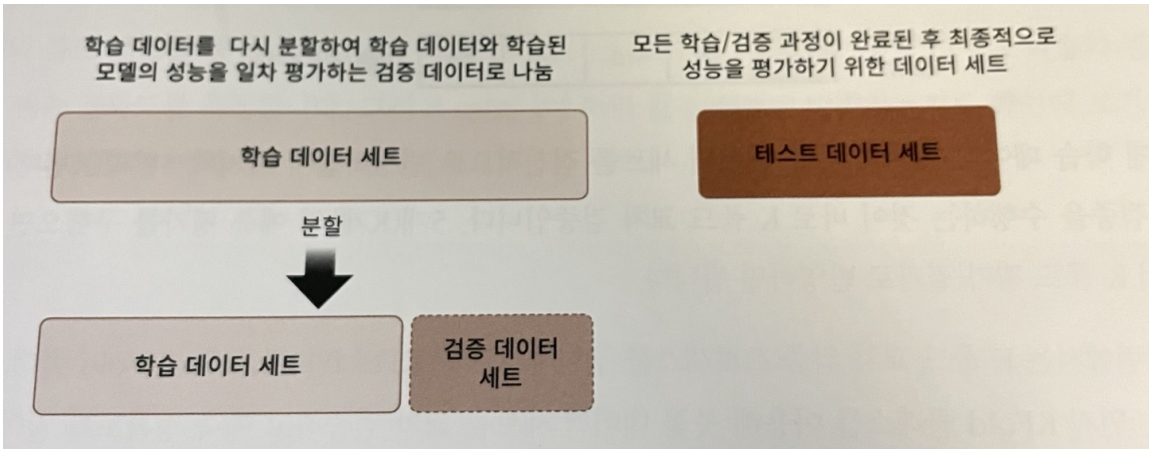
- 사이킷런 내장 데이터 : iris, boston, breast_cancer ,,,

모듈

1. train_test_split

- 파라미터 : test size, shuffle(디폴트 T), random_state 등
- 반환값 튜플 형태

2. 교차 검증 : 데이터 편중 막기 위해 별도의 여러 세트로 구성된 학습 데이터와 검증 데이터 세트에서 학습과 평가 수행/ 이에 따라 하이퍼파라미터 튜닝 등의 모델 최적화를 쉽게 가능
   과적합: 모델이 학습 데이터에만 과도하게 최적화되어 실제 예측을 못하는 경우(성능 저하)



3. k 폴드 교차 검증 : k개의 데이터 폴드 세트를 만들어 k번 만큼 각 세트에 학습+평가 반복
4. Stratified k 폴드 : 불균형한 분포를 가진 레이블 데이터 집합 (특정 레이블 값이 많거나 적은)
   원본 데이터 레이블 분포 고려 후 이와 동일하게 학습/검증 데이터 분배
   왜곡된 레이블 데이터에서 꼭 사용! 회귀에선 불가! 숫자니까!
5. cross_val_score
   - cv : 교차검증 폴드 수, scoring : 예측 성능 평가 지표
   - cross_validate는 평가지표 여러 개 가능
6. GridSearchCV : 교차검증 + 최적 하이퍼 파라미터 튜닝

   max_depth * min_samples_split 번 만큼 수행하면서 최적 찾음

데이터 전처리

1) 데이터 인코딩

- 레이블 인코딩 : 카테고리 피처를 코드형 숫자 값으로 변환
  하지만, 숫자의 크고 작음의 특성이 작용하기 때문에 변환값을 중요도로 인식하면 안 됨.
- 원핫 인코딩 : 새로운 피처를 추가해 고유값 해당하는 칼럼만 1 표시하는 것
  - 주의!! **모든 문자열 값을 숫자형 값으로 변환**해서 진행, 입력값 2차원 데이터 필요
  - pd.get_dummies로도 가능

2) 피처 스케일링과 정규화

- 표준화 : 평균 0, 표준편차 1인 가우시안 정규 분포로 변환
  - **StandardScaler**
    → 예측 성능 향상 도움 (SVM, 선형회귀, 로지스틱 회귀 등에서 특히)
  - **MinMaxScaler**
    → 데이터 분포가 가우시안 분포 아닐 때, 0~1 사이 값으로 변환
- 정규화 : 피처 크기 통일
  X_new= (X-min(X)) / (max(X)-min(X))

*** train, test 스케일링 할 때, 두 데이터의 서로 다른 원본값이 동일 값으로 변환되는 문제 발생 가능하므로 테스트 데이터는 반드시 학습 데이터 스케일링 기준에 따라야함. test : fit 적용 x

전체 대상 스케일하고 학습,테스트 데이터로 분리하는 게 좋음!

타이타닉 생존자 예측

1) 성별, 나이, 티켓 등급에 따른 생존 확률 보기

2) 나머지 문자열 카테고리 피처 > 숫자형으로 변환 : 레이블 인코딩

3) 모델 학습

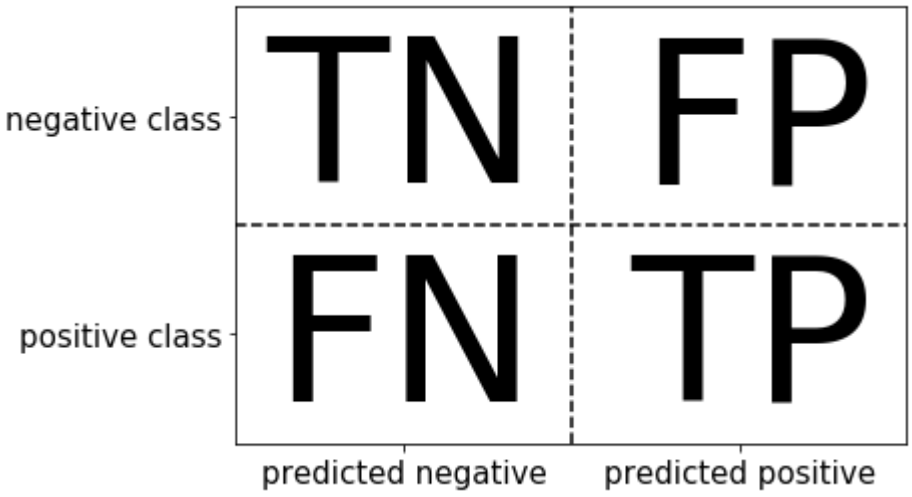
4) 예측

CH3

평가 성능 지표

1. 분류

- 정확도
  - 예측 결과 동일한 데이터 건수/전체 예측 데이터 건수
  - 직관적 예측 성능 나타냄
  - 이진 분류의 경우 왜곡 가능성 있음
  - 불균형 레이블 값 분포에서 적합한 지표 아님
- 오차 행렬
  - 실제와 예측 레이블 클래스 값이 어떻게 다른지



** 정확도=(TN+TP)/(TN+FP+FN+TP)인데 불균등 분포에선 neg가 많아 neg 예측 정확도가 높아지는 문제

- 정밀도
  TP/(TP+FP) = 진짜 양성/예측 양성
  실제 음성 데이터를 양성으로 잘못 판단 시 큰 문제, FP 낮추는 데 초점

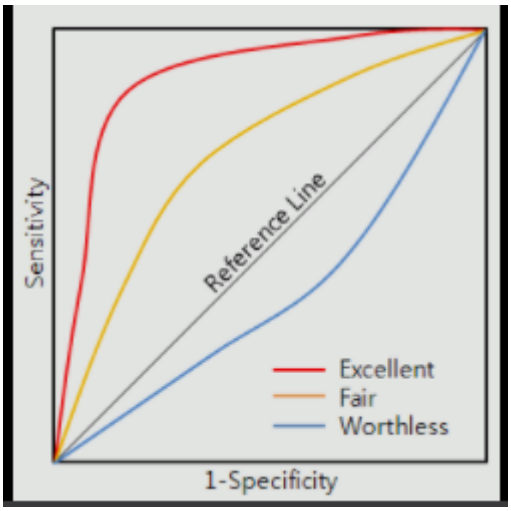
- 재현율
  TP/(TP+FN) = 진짜 양성/ 실제 양성 =민감도= 진짜 양성 비율(TPR)

  실제 양성 데이터를 음성으로 잘못 판단 시 큰 문제가 되기 때문에 중요함!!!, FN 낮추는 데 초점

→ 보완적인 역할, 둘 다 높아야 좋음

→ 정밀도 재현율 트레이드 오프 : predict_proba
EX) 임계값을 낮추니 재현율 증가, 정밀도 감소 : POSITIVE 예측 더 널널하게 하니까

- F1 스코어 : f1_score
  정밀도와 재현율 결합 지표(조화 평균)
- ROC : FPR(가짜 양성 비율)에 대한 TPR(재현율)의 곡선 : roc_curve
- AUC : FPR 변할 때 TPR 변화 곡선



** 재현율 = 민감도 =TPR

** 특이성= TNR =TN/(FP+TN)

** FPR=예측양성/실제양성=1-(진짜음성/예측음성)= 1- TNR= 1- 특이성

→ ROC곡선은 민감도에 대한 1-특이도 곡선

피마 인디언 당뇨병 예측

1) 각 평가지표 구하기

2) 전체 데이터의 64%(f1)가 음성이므로 재현율 성능 초점

3) 그래프 보니 임계값이 0.42정도면 정밀도와 재현율 균형점 하지만 지표 수치가 낮음 → 뭐가 문제일까? 전체 데이터 다시 보기

4) 0 값이 많은 변수는 훈련에 방해되므로 찾아서 평균 대체

5) 피처 스케일링

6) 임계값 변화하면서 재현율 보기