EURON 1주차 과제_2장

사이킷런으로 시작하는 머신러닝

사이킷런 소개와 특징

- 가장 많이 사용되는 라이브러리
- 가장 파이썬스러운 API 제공
- 머신러닝을 위한 매우 다양한 알고리즘과 개발을 위한 편리한 프레임워크와 API 제공

첫번째 머신러닝 만들어 보기 - 붓꽃 품종 예측하기

1. 지도학습: 명확한 정답이 주어진 데이터를 먼저 학습한 뒤 미지의 정답을 예측하는 방식
2. 레이블(Label) 데이터: 분류 결정값
3. 하이퍼 파라미터: 머신러닝 알고리즘별 최적의 학습을 위해 직접 입력하는 파라미터들
   이를 통해 머신러닝 알고리즘의 성능을 조정(튜닝)할 수 있음

실습은 ipynb 파일 참고

사이킷런의 기반 프레임워크 익히기

Estimator 이해 및 fit(), predict() 메서드

- fit() : ML 모델 학습
- predict() : 학습된 모델의 예측
- 지도학습의 주요 두 축 : 분류(Classification), 회귀(Regression)
- Classifer : 분류 알고리즘 구현한 클래스
- Regressor : 회귀 알고리즘 구현한 클래스
- Classifier와 Regressor를 합쳐 Estimator 클래스라고 부름
- Estimator은 지도학습의 모든 알고리즘 종류의 집합

사이킷런의 주요 모듈

교재 92pg 참고

내장된 예제 데이터 세트

교재 94pg 참고

- 데이터 타입은 일반적으로 딕셔너리 형태
- 키는 보통 data, target, target_name, feature_names, DESCR로 구성
- data와 target은 넘파이 배열 (ndarray)
- target_names, feature_names는 넘파이 배열 또는 파이썬 리스트 타입
- DESCR은 스트링 타입

Model Selection 모듈 소개

학습/테스트 데이터 세트 분리 - train_test_split()

- 학습 데이터 세트를 기반으로 예측 시 정확도 100% → 데이터 세트 분리해야함
- sklearn.model_selection 모듈에서 train_test_split. 로드
- train_test_split()는 첫번째 파라미터로 피처 데이터 세트, 두 번째 파라미터로 레이블 데이터 세트 입력

교차 검증

- 과적합(Overfitting) : 모델이 학습 데이터에만 과도하게 최적화되어 실제 예측을 다른 데이터로 수행할 경우 예측 성능이 과도하게 떨어지는 것 → 개선을 위해 교차 검증 이용
- K 폴드 교차 검증 : 가장 보편적으로 사용, K개의 데이터 폴드 세트를 만들어 K번만큼 각 폴드 세트에 학습과 검증 평가를 반복적으로 수행
- Stratified K 폴드 : 불균형한 분포도를 가진 레이블 데이터 집합을 위한 K 폴드 방식
- 분류에서의 교차 검증은 Strarified K 폴드
- 회귀에서는 Stratified K 폴드 지원 안됨

교차 검증을 보다 간편하게 - cross_val_score()

- 교차 검증을 편리하게 수행할 수 있게 해주는 API

- cross_validate() : 유사한 API, 여러 개의 평가 지표 반환 가능

GridSearchCV - 교차 검증과 최적 하이퍼 파라미터 튜닝을 한번에

- 하이퍼 파라미터 : 머신러닝 알고리즘을 구성하는 주요 구성 요소, 값을 조정해 알고리즘의 예측 성능 개선
- 여러 종류의 하이퍼 파라미터를 다양하게 테스트하여 최적의 파라미터 편리하게 찾게 해줌
- 수행시간이 생대적으로 오래 걸림

데이터 전처리

- Garbage in, garbage out
- 결손값(NaN, Null)은 허용하지 않음
- 피처 값 중 null이 얼마 되지 않는다면 피처의 평균값으로 대체 가능
- 사이킷런 패키지는 문자열 값을 입력값으로 허용하지 않음
- 모든 문자열 값은 인코딩돼서 숫자 형으로 변환시켜야 함

데이터 인코딩

▼ 레이블 인코딩

- 카테고리 피처를 코드형 숫자 값으로 변환
- LabelEncoder 클래스로 구현
- 레이블 인코딩은 선형 회귀와 같은 ML 알고리즘에는 적용하지 않음(트리 계열의 알고리즘은 상관X)

▼ 원-핫 인코딩

- 피처 값의 유형에 따라 새로운 피처를 추가해 고유 값에 해당하는 칼럼에만 1을 표시하고 나머지 칼럼에는 0을 표시하는 방법
- OneHotEncoder 클래스로 변환 가능
- 입력값으로 2차원 데이터 필요
- 변환값이 희소 행렬 형태이므로 이를 다시 toarray() 메서드를 이용해 밀접 행렬로 변환해야함

피처 스케일링과 정규화

- 서로 다른 변수의 값 범위를 일정한 수준으로 맞추는 작업
- 대표적인 방법으로 표준화와 정규화

StandardScaler

- 표준화를 쉽게 지원하기 위한 클래스
- 개별 피처를 평균이 0이고 분산이 1인 값으로 변환 → RBF 커널을 이용하는 서포트 벡터 머신이나 선형 회귀, 로지스틱 회귀에서 에측 성능 향상에 중요한 요소

MinMaxScaler

- 데이터값을 0과 1 사이의 범위 값으로 변환
- 데이터 분포가 가우시안 분포가 아닐 경우 Min, Max Scale 적용