Chapter 1. 파이썬 기반의 머신러닝과 생태계 이해

1-1. 머신 러닝의 개념

머신러닝 - 어플리케이션을 수정하지 않고도 데이터를 기반으로 패턴을 학습하고 결과를 예측하는 알고리즘 기법

분류 – 지도학습(supervised learning), 비지도 학습(unsupervised learning), 강화학습(reinforcement learning)

- 지도학습: 분류, 회귀, 추천 시스템, 시각/음성 감지/인지, 텍스트 분석, NLP
- 비지도학습: 클러스터링, 차원축소, 강화학습

1-2. 파이썬 머신러닝 생태계를 구성하는 주요 패키지

머신러닝 패키지-사이킷런

행렬/선형대수/통계 패키지: 넘파이, 사이파이

데이터 핸들링: 판다스, matplotlib

시각화: matplotlib, seaborn

1-3. 넘파이

배열 기반의 연산, 데이터 핸들링 기능

1-4. 데이터 핸들링 – 판다스

행과 열로 이뤄진 2차원 데이터를 효율적으로 가공, 처리할 수 있는 기능 제공

파이썬의 리스트, 컬렉션, 넘파이 등의 내부 데이터뿐만 아니라 csv 등의 파일을 쉽게 데이터프레임으로 변경해 데이터의 가공 분석을 편리하게 수행할 수 있게 만들어줌.

핵심 객체 : DataFrame – 여러 개의 행과 열로 이뤄진 2차원 데이터를 담는 데이터 구조체

Chapter 2. 사이킷런으로 시작하는 머신러닝

2-1. 사이킷런 소개와 특징

사이킷런 – 파이썬 머신러닝 라이브러리 중 가장 많이 사용되는 라이브러리

- 파이썬 기반의 다른 머신러닝 패키지도 사이킷런 스타일의 API 지향할 정도로 쉽고 파이썬스러운 API 제공
- 머신러닝을 위한 매우 다양한 알고리즘과 개발을 위한 편리한 프레임워크와 API 제공
- 오랜 기간 실전 환경에서 검증됨. 매우 많은 환경에서 사용되는 성숙한 라이브러리

2-2. 첫 번째 머신러닝 만들어 보기 - 붓꽃 품종 예측하기

붓꽃 데이터 세트로 붓꽃 품종 분류(classification)하기

분류 - 대표적인 지도학습 방법 중 하나

=> 학습을 위한 다양한 피처와 분류 결정값인 레이블 데이터로 모델 학습한 뒤 별도의 테스트 데이터 세트에서 미지의 레이블 예측. 명확한 정답이 주어진 데이터 세트를 먼저 학습한 뒤 미지의 정답을 예측하는 방식.

- sklearn.datasets 내의 모듈: 사이킷런에서 자체적으로 제공하는 데이터 세트를 생성하는 모듈의 모임

- sklearn.tree 니의 모듈: 트리 기반 ML 알고리즘을 구현한 클래스의 모임
- sklearn.model_selection: 학습 데이터와 검증 데이터, 예측 데이터로 데이터를 분리하거나 최적의 하이퍼 파라미터로 평가하기 위한 다양한 모듈의 모임
- 하이퍼 파라미터: 머신러닝 알고리즘 별로 최적의 학습을 위해 직접 입력하는 파라미터들 통칭. 머신러닝 알고리즘의 성능 튜닝

2-3. 사이킷런의 기반 프레임워크 익히기

Estimator 이해 및 fit(), predict() 메서드
- fit(): ML 모델 학습
- predict(): 학습된 모델의 예측
- Estimator 클래스: Classifier 과 Regressor 을 통칭. 지도학습의 모든 알고리즘을 구현한 클래스
- 비지도학습(차원 축소, 클러스터링, 피처 추출): 대부분 fit(), transform() 적용
  - fit(): 입력 데이터의 형태에 맞춰 데이터를 변환하기 위한 사전 구조를 맞추는 작업
  - transform(): 입력 데이터의 차원 변환, 클러스터링, 피처 추출 등 실제 작업 수행

사이킷런의 주요 모듈
- 예제 데이터: sklearn.datasets
- 피처 처리: sklearn.preprocessing, sklearn.feature_selection, sklearn.feature_extraction
- 피처 처리 & 차원 축소: sklearn.decomposition
- 데이터 분리, 검증 & 파라미터 튜닝: sklearn.model_selection
- 평가: sklearn.metrics
- ML 알고리즘: sklearn.ensemble, sklearn.linear_model, sklearn.naive_bayes, sklearn.neighbors, sklearn.svm, sklearn.tree, sklearn.cluster
- 유틸리티: sklearn.pipeline

주요 프로세스-
피처의 가공 변경 추출을 수행하는 피처 처리, ML 알고리즘 학습 예측 수행, 모델 평가의 단계 반복 수행

내장된 예제 데이터 세트
- 회귀 용도: datasets.load_boston(), datasets.load_diabetes()
- 분류 용도: datasets.load_breaset_cancer(), datasets.load_digits(), datasets.load_iris()
- fetch 계열 명령: 데이터 크기가 커 패키지에 처음부터 저장되어 있지 않고 인터넷에서 내려받아 서브 디렉터리에 저장 후 추후 불러들이는 데이터
- 일반적으로 딕셔너리 형태로 되어 있음.
- 키: data, target, target_name, feature_names, DESCR 로 구성
  - data: 피처의 데이터 세트
  - target: 분류 시 레이블 값, 회귀일 때는 숫자 결괏값 데이터 세트
  - target_names: 개별 레이블의 이름
  - feature_names: 피처의 이름
  - DESCR: 데이터 세트에 대한 설명과 각 피처의 설명

2-4. Model Selection 모듈 소개

model_selection 모듈: 학습 데이터와 테스트 데이터 세트를 분리하거나 교차 검증 분할 및 평가, Estimator 의 하이퍼 파라미터를 튜닝하기 위한 다양한 함수와 클래스 제공

학습/테스트 데이터 세트 분리 - train_test_split()

train_test_split(피처 데이터 세트, 레이블 데이터 세트)

- test_size: 전체 데이터에서 테스트 데이터 세트 크기를 얼마로 샘플링할 것인가 결정
- train_size: 전체 데이터에서 학습용 데이터 세트 크기를 얼마로 샘플링할 것인가 결정
- shuffle: 데이터를 분리하기 전 데이터를 미리 섞을지를 결정. 디폴트=True. 데이터를 분산시켜 더 효율적인 학습 및 테스트 데이터 세트를 만드는 데 사용
- random_state: 호출할 때마다 동일한 학습/테스트용 데이터 세트를 생성하기 위해 주어지는 난수 값.
- train_test_split() 반환값은 튜플 형태. 순차적으로 학습용 데이터의 피처 데이터 세트, 테스트용 데이터의 피처 데이터 세트, 학습용 데이터의 레이블 데이터 세트, 테스트용 데이터의 레이블 데이터 세트 반환

교차 검증

과적합 개선 위해 교차 검증을 이요해 더 다양한 학습과 평가 수행

- 과적합: 모델이 학습 데이터에만 과도하게 최적화되어 실제 예측을 다른 데이터로 수행할 경우 예측 성능이 과도하게 떨어지는 것
- 교차 검증: 데이터 편중을 막기 위해 별도의 여러 세트로 구성된 학습 데이터 세트와 검증 데이터 세트에서 학습과 평가 수행 => 수행한 평가 결과에 따라 하이퍼 파라미터 튜닝 등의 모델 최적화
- ML 모델의 성능 평가: 교차 검증 깁나으로 1 차 평가 후 최종적으로 테스트 데이터 세트에 적용해 평가

K 폴드 교차 검증

- 가장 보편적으로 사용되는 교차 검증 기법. K 개의 데이터 폴드 세트를 만들어 K 번만큼 각 폴드 세트에 학습과 검증 평가를 반복적으로 수행하는 방법.
- KFold 와 StartifiedKFold 클래스 제공

Stratified K 폴드

- 불균형한 분포도를 가진 레이블 데이터 집합을 위한 K 폴드방식
- K 폴드가 레이블 데이터 집합이 원본 데이터 집합의 레이블 분포를 학습 및 테스트 세트에 제대로 분배하지 못하는 경우의 문제 해결

교차 검증을 보다 간편하게 - cross_val_score()

- 폴드 세트를 설정
- for 루프에서 반복으로 학습 및 테스트 데이터의 인덱스 추출
- 반복적으로 학습과 예측 수행 에측 성능 반환

cross_val_score(estimator, X, y=None, scoring=None, cv=None, n_jobs=1, verbose=0, fit_params=None,pre_dispatch='2*n_jobs')
- estimator: Classifier 또는 Regressor
- X: 피처 데이터 세트
- y: 레이블 데이터 세트
- scoring: 예측 성능 평가 지표 기술
- cv: 교차 검증 폴드 수
GridSearchCV - 교차 검증과 최적 하이퍼 파라미터 튜닝을 한 번에
하이퍼 파라미터- 머신러닝 알고리즘을 구성하는 주요 구성 요소. 이 값을 조정해 알고리즘의 예측 성능 개선
파라미터의 집합을 만들고 순차적으로 적용하면서 최적화 수행 가능
GridSearchCV- 교차 검증을 기반으로 하이퍼 파라미터의 최적 값을 찾게 해줌.
- estimator: classifier, regressor, pipeline
- param_grid: key +리스트 값을 가지는 딕셔너리가 주어짐. estimator 의 튜닝을 취해 파라미터명과 사용될 여러 파라미터 값 지정
- scoring: 예측 성능을 측정할 평가 방법 지정. 사이킷런의 성능 평가 지표를 지정하는 문자열로 지정하나 별도의 성능 평가 지표 함수도 지정 가능
- cv: 교차 검증을 위해 분할되는 학습, 테스트 세트의 개수 지정
- refit: 디폴트 True. True 로 생성 시 가장 최적의 하이퍼 파라미터를 찾은 뒤 입력된 estimator 객체를 해당 하이퍼 파라미터로 재학습시킴.
2-5. 데이터 전처리
결손값 처리, 카테고리형 피처 텍스트 피처 숫자형 변환
데이터 인코딩
- 레이블 인코딩: 카테고리 피처를 코드형 숫자 값으로 변환
  - LabelEncoder 클래스로 구현
- 원핫 인코딩 : 피처 값의 유형에 따라 새로운 피처 추가해 고유 값에 해당하는 칼럼에만 1 표시 나머지 칼럼에는 0 표시
  - OneHotEncoder 클래스로 변환 가능. 변환 전 모든 문자열 값이 숫자형 값으로 변환되어야 함. 입력 값으로 2 차원 데이터 필요
피처 스케일링과 정규화
- 피처 스케일링: 서로 다른 변수의 값 범위를 일정한 수준으로 맞추는 작업
- 표준화: 데이터의 피쳐 각각의 평균 0 분산 1 인 가우시안 정규 분포를 가진 값으로 변환
- 정규화: 서로 다른 피처의 크기 통일 위해 크기 변환
MinMaxScaler
- 데이터 값을 0 과 1 사이 범위 값으로 변환.
- 데이터 분포가 가우시안 분포가 아닐 경우 min, max scale 적용 가능
학습 데이터와 테스트 데이터의 스케일링 변환 시 유의점

- fit(): 데이터 변환을 위한 기준 정보 설정 적용
- transform(): 설정된 정보를 이용해 데이터 반환

유의할 점

1. 가능하다면 전체 데이터의 스케일링 변환을 적용한 뒤 학습과 테스트 데이터로 분리
2. 1 이 여의치 않다면 테스트 데이터 변환 시에는 fit()이나 fit_transform()을 적용하지 않고 학습 데이터로 이미 fit()된 Scaler 객체를 이용해 transform()으로 변환

2-6. 사이킷런으로 수행하는 타이타닉 생존자 예측

Chapter 3. 평가

성능 평가 지표

- 회귀: 대부분 실제값과 예측값의 오차 평균값
- 분류: 실제 결과 데이터와 예측 결과 데이터가 얼마나 정확하고 오류가 적게 발생하는지
  - 정확도, 오차행렬, 정밀도, 재현율, F1 스코어, ROC AUC
  - 이진 분류: 긍정 부정 같은 2 개 결괏값만 가짐
  - 멀티 분류: 여러 개의 결정 클래스 값을 가짐

3-1. 정확도(Accuracy)

직관적으로 모델 예측 성능을 나타내는 평가 지표

이진 분류의 경우 데이터의 구성에 따라 ML 모델 성능을 왜곡할 수도 있음

3-2. 오차 행렬(confusion matrix)

이진 분류의 예측 오차가 얼마인지, 어떤 유형의 예측 오류가 발생하고 있는지 나타내는 지료

- TN: 예측값을 negative 값 0 로 예측했는데 실제 값 역시 negative 값 0
- FP: 예측값을 positive 값 1 로 예측했는데 실제 값은 negative 값 0
- FN: 예측값을 negative 값 0 로 예측했는데 실제 값은 positive 값 1
- TP: 예측값을 positive 값 1 로 예측했는데 실제 값 역시 positve 값 1

3-3. 정밀도와 재현율

- 정밀도 = TP / (FP + TP)
  - 실제 negative 음성인 데이터 예측을 positive 양성으로 잘못 판단하게 되면 큰 영향이 발생하는 경우
  - ex) 스팸메일 여부 판단 모델
- 재현율 = TP / (FN + TP)
  - 실제 positive 양성 데이터를 negative 로 잘못 판단하면 큰 영향이 발생하는 경우
  - ex) 암 판단 모델, 금융 사기 적발 모델

정밀도 / 재현율 트레이드 오프

- 정밀도와 재현율은 상호 보완적인 평가 지표이므로 어느 한쪽을 강제로 높이면 다른 하나의 수치가 떨어지기 쉬움
- predict_proba(): 테스트 피처 데이터 세트를 파라미터로 입력해주면 테스트 피처 레코드의 개별 클래스 예측 확률 반환

3-4. F1 스코어

정밀도와 재현율을 결합한 지표

- 정밀도와 재현율이 어느 한 쪽으로 치우치지 않는 수치를 나타낼 때 상대적으로 높은 값 가짐

3-5. ROC 곡선과 AUC

- ROC 곡선: 수신자 판단 곡선. 이진 분류 모델의 예측 성능 판단하는 평가 지표
  - fpr 을 x 축으로, tpr 을 y 축으로 잡으면 곡선 형태로 나타남.
  - 민감도(tpr): 실제값 positive 가 정확히 예측돼야 하는 수준
  - 특이성(tnr): 실제값 negative 가 정확히 예측돼야 하는 수준

3-6. 피마 인디언 당뇨병 예측

당뇨병 여부 판단 머신러닝 예측 모델 수립, 평가 지표 적용