EURON_1주차 STUDY

chapter 1 : 파이썬 기반의 머신러닝과 생태계 이해

01. 머신러닝의 개념

→ 데이터를 기반으로 숨겨진 패턴을 인지해 해결함. 머신러닝 알고리즘은 데이터를 기반으로 통계적인 신뢰도를 강화하고 예측 오류를 최소화하기 위한 다양한 수학적 기법을 적용

머신러닝의 분류

지도학습 : 분류, 회귀, 추천 시스템, 시각/음성 감지/인지, 텍스트 분석

비지도 학습 : 클러스터링, 차원축소, 강화학습

데이터 전쟁

머신러닝의 단점 → 데이터 의존적. 따라서 데이터를 이해하고 효율적으로 가공, 처리, 추출해 최적의 데이터를 기반으로 알고리즘을 구동할 수 있도록 준비

파이썬과 R 기반 머신러닝 비교

파이썬이 R에 비해 뛰어난 점 : 파이썬은 소리없이 프로그램이 세계를 점령하고 있는 언어

02. 파이썬 머신런이 생태계를 구성하는 주요 패키지

→ 머신러닝 패키지, 행렬/선형대수/통계 패키지, 데이터 핸들링, 시각화

03. 넘파이

넘파이는 배열 기반의 연산은 물론이고 다양한 데이터 핸들링 기능을 제공. 많은 파이썬 기반의 패키지가 넘파이를 이용해 데이터 처리

넘파이 ndarry 개요

넘파이에서 다차원 배열을 쉽게 생성하고 다양한 연산을 수행할 수 있음.

ndarray의 데이터 타입

숫자값, 문자열 값, 불값 모두 가능하다. 만약 타입이 다르다면 더 큰 데이터 타입으로 형변환 일괄 적용

ndarray를 편리하게 생성하기

arange : array를 range로 표현

zeros : 함수 인자로 튜플 형태의 shape값을 입력하면 모든 값을 0으로 채운 해당 shape를 가진 ndarray로 변환

ones : 모든 값을 1로 채움

ndarray의 차원과 크기를 변경하는 reshape()

ndarray를 특정 차원 및 크기로 변환한다.

넘파이의 ndarray의 데이터 세트 선택하기 - 인덱싱(indexing)

일부 데이터 세트나 특정 데이터만을 선택할 수 있도록 하는 인덱싱

특정한 데이터만 추출, 슬라이싱, 팬시 인덱싱, 불린 인덱싱

행렬의 정렬 - sort()와 argsort()

sort : 원 행렬 자체를 정렬한 형태로 변환하며 변환 값은 없음

argsort : 정렬 행렬의 원본 행렬 인덱스를 ndarray형으로 반환

선형대수 연산 - 행렬 내적과 전치 행렬 구하기

행렬 내적 : np.dot()

전치 행렬 : np.transpose()

04. 데이터 핸들링 - 판다스

판다스는 데이터 처리를 위해 존재하는 가장 인기 라이브러리

info : 총 데이터 건수와 데이터 타입, null 건수를 알 수 있음

describe : 칼럼별 숫자형 데이터값의 n-percentile분포도, 평균값, 최댓값, 최솟값나타냄

value_counts : 해당 칼럼값의 유형과 건수를 확인

series : index와 단 하나의 칼럼으로 구성된 데이터 세트

DataFrame과 리스트, 딕셔너리, 넘파이 ndarray 상호변환

** ipynb 코드참고

dataframe 데이터 삭제

drop 이용

칼럼을 삭제하는 경우, axis =1 지정

index 객체

고유하게 식별할 수 있는 객체.

df.index or series.index이렇게 사용

단일 값 반환 및 슬라이싱도 가능하다

reset_index : 새롭게 인덱스를 다시 만든다.

데이터 셀렉션 및 필터링

ix, iloc, loc 연산자를 통해 작업을 수행 특히 []연산자 사용을 주의해야한다.

정렬, aggregation, groupby 적용

sort_values : 데이터의 정렬

aggregation : min, max, sum, count등 사용가능

groupby : 그룹을 기준으로 연산 시행

결손 데이터 처리하기

isna : 결손 데이터 여부확인

fillna : 다른값 대체

apply lambda식으로 데이터 가공

파이썬 프로그래밍을 지원하기 위해 만들어짐.

복잡한 데이터 가공이 필요할 때 사용

chapter 2 : 사이킷런으로 시작하는 머신러닝

사이킷런 소개와 특징

머신러닝을 위한 매우 다양한 알고리즘 개발을 위한 편리한 프레임워크와 api 제공

오랜 기간 실전 환경에서 검증됐으며, 매우 많은 환경에서 사용되는 라이브러리

첫번째 머신러닝 만들어보기

데이터를 분리 → 객체 생성(decisiontreeclassifier) → 학습 수행 → 예측 수행 → 예측 정확도

사이킷런의 기반 프레임워크 익히기

estimator 이해 및 fit(), predict()메서드

fit은 학습, predict는 예측

model selection 모듈 소개

train_test_split : 학습/테스트 데이터 세트 분리

교차검증

교차검증은 본고사를 치르기 전에 모의고사를 여러 번 보는 것과 같다. 데이터 편중을 막기 위해서 별도의 여러 세트로 구성된 학습 데이터 세트와 검증 데이터 세트에서 학습과 평가를 수행하는 것. 그리고 각 세트에서 수행한 평가 결과에 따라 하이퍼 파라미터 뉴팅 등의 모델 최적화를 손쉽게 할 수 있다.

k 폴드 교차 검증

먼저 k개의 데이터 폴드 세트를 만들어서 k번만큼 각 폴드 세트에학습과 검증 평가를 반복적으로 수행

stratified k 폴드

불균형한 분포도를 가진 레이블 데이터 집합을 위한 k폴드 방식. 레이블 데이터 집합이 원본 데이터 집합의 레이블 분포를 학습 및 테스트 세트에 제대로 분배하지 못하는 경우의 문제 해결

gridsearchCV : 교차검증과 최적 하이퍼 파라미터 튜닝을 한 번에

알고리즘에 사용되는 하이퍼 파라미터를 순차적으로 입력하면서 편리하게 최적의 파라미터를 도출할 수 있음

데이터 전처리

데이터 인코딩

레이블 인코딩 : 카테고리 피처를 코드형 숫자 값으로 변환

원-핫 인코딩 : 피처 값의 유형에 따라 새로운 피처를 추가해 고유 값에 해당하는 칼럼에만 1을 표시

피처 스케일링과 정규화

표준화와 정규화가 있음

StandardScaler

표준화를 쉽게 지원하기 위한 클래스

minmaxscaler

데이터를 0과 1 사이의 범위 값으로 변환

학습데이터와 테스트 데이터의 스케일링 변환 시 유의점은 학습데이터를 다시 transform변환을 시킨 후 테스트 데이터에 적용을 시켜야 한다는 점이다.

머신러닝 모델은 학습 데이터를 기반으로 학습되기 때문에 반드시 테스트 데이터는 학습 데이터의 스케일링 기준에 따라야 함

사이킷런으로 수행하는 타이타닉 생존자 예측

**ipynb코드참고

chapter 3: 평가

성능 평가 지표는

정확도, 오차행렬, 정밀도, 재현율, F1 스코어, ROC ACU 등이 있다.

분류는 이진분류와 멀티 분류로 나눌 수 있다.

01. 정확도 (Accuracy)

정확도는 실제 데이터에서 예측 데이터가 얼마나 같은지 판단하는 지표

정확도 = 예측 결과가 동일한 데이터 건수/전체 예측 데이터 건수

02. 오차행렬

이진 분류의 예측 오류가 얼마인지와 더불어 어떠한 유형의 예측 오류가 발생하고 있는지를 함께 나타내는 지표

TN, FP, FN, TP

03. 정밀도와 재현율

정밀도 = TP/(PF+TP)

재현율 = TP/(FN+TP)

정밀도는 예측을 P로 한 대상 중에 예측이 실제 P로 나타난 데이터의 비율이다.

재현율은 실제값이 P인 대상중에서 예측과 실제 값이 P로 일치한 데이터의 비율. 민감도라고도 불린다.

재현율이 상대적으로 더 중요한 지표인 경우는 실제 양성인 데이터 예측을 음성으로 잘못 판단하게 되면 업무상 큰 영향이 발생하는 경우

정밀도가 상대적으로 더 중요한 지표인 경우는 실제 음성인 데이터를 양성으로 잘못 판단하게 되면 업무상 큰 영향이 발생하는 경우

정밀도/재현율 트레이드오프

정밀도와 재현율은 상호 보완적인 평가 지표

시각화를 통해 이를 확인할 수 있다.

04. F1스코어

F1 =2*(정밀도*재현율/정밀도+재현율)

정밀도와 재현율이 어느 한쪽으로 치우치지 않는 수치를 나타낼 때 상대적으로 높은 값을 가진다.

05. ROC곡선과 AUC

ROC곡선 : 수신자 판단 곡선, FRP이 변할때 TPR이 어떻게 변하는지 나타내는 곡선

FPR = 1 - 특이성

06. 피마 인디언 당뇨병 예측

** ipynb 코드 확인