21.02.25 머신러닝

노트북: [면접 관련]
만든 날짜: 2021-02-25 오후 6:49 **수정한 날짜:** 2021-04-20 오전 1:42
작성자: 황인범

---

1. 머신러닝 / 딥러닝 지식
    1. 배경지식
        1. 머신러닝 이란 ? 애플리케이션을 수정하지 않고 데이터를 통해 패턴을 학습하여 결과를 예측하는 알고리즘 기법 / 데이터를 기반으로 숨겨진 패턴을 인지
        2. 머신러닝의 분류
            1. 지도 학습
                1. 분류
                2. 회귀
                3. 추천시스템
                4. 시각/음성, 감지/인지
                5. 텍스트분석, NLP
            2. 비지도 학습
                1. 군집화(클러스터링)
                2. 차원축소
                3. 강화학습
        3. 알고리즘 < 데이터 -> 둘다 중요하지만 데이터가 더 중요하다 - 가비지 인 가비지 아웃
        4. 파이썬의 장점
            1. 높은 개발 생산성
            2. 오픈 소스계열의 지원을 받으며 많은 라이브러리가 있다
            3. 인터프리터 언어의 특징 - 느리지만 뛰어난 확장성, 유연성, 호환성 -> 서버, 네트워크, 시스템, IOT
            4. 머신러닝 애플리케이션과 결합해 애플리케이션 개발이 가능하다.
            5. 기업환경으로 확산이 가능하다.
            6. 텐서플로, 케라스,파이토치등이 가능하다
        5. 행렬/선형대수/통계 패키지 - 넘파이
        6. 데이터 핸들링 - 판다스
            1. iloc 위치기반
            2. loc 명칭기반
            3. sort_value(by=['컬럼'], ascending=False) / .sort()
            4. groupby('컬럼명').어쩌고
            5. agg(['컬럼명','컬럼명']) - 여러 컬럼을 지정
            6. .isna() - 널값 확인
            7. fillna('???') - 널값 대체
            8. apply lambda - squares = map(lambda x : x **2, a) / .apply(lambda x : len(x)) - 칼럼에 일괄적으로 데이터 가공
        7. 넘파이
    2. 사이킷런
        1. train_test_split(data, target, testsize=0.3, random_state=777)
        2. 교차검증 - 과적합(모델이 현재 학습 데이터에만 과도학게 최적화되어 실제 예측을 다른 데이터로 수행하는 경우 성능이 떨어지는 경우)
            1. k-폴드 - k개의 데이터 폴드 세트를 만들어 수행
            2. stratified k 폴드 - 불균형한 분포도를 가진 레이블을 위한 k폴드 방식으로 일정이상의 레이블을 유지
            3. loocv - 데이터가 적을때 사용하는 것으로 레이블을 하나로 나머지를 학습으로 사용하는 방법
            4. hold out - 내가 임의로 데이터 셋을 쪼개는 것

5. cross_val_score() - 모델 + 교차검증 + 성능평가를 한번에 하는 방법

3. 하이퍼 파리미터
    1. gridSearchCV - 교차검증과 최적 하이퍼 파라미터 튜닝. 구간 내에 몇가지 값 넣고 찾는 것
    2. Manual Search - 직접 넣어보는거
    3. Random Search - 구간내 값을 랜덤 샘플링을 통해 선정
    4. Bayesian optimization - 알려지지 않는 목적함수를 최대(최소)로 하는 최적해 기법
        1. surrogate 모델 - 입출력 특성을 실제모형과 유사하게 만드는 것을 목적 / 추상화된 모델을 통해서 입력과 출력의 관계를 설명 (가우시안 프로세스 많이씀)
        2. Acquistion(어큐션) 함수 - 확률적 추정 결과를 바탕으로 다음 입력값 후보를 추천한다.

3. 데이터 전처리
    1. 데이터 인코딩
        1. 레이블 인코딩 - 카테고리 피처를 코드형 숫자 값으로 변환하는 것
        2. 원핫 인코딩 - 피처값의 유형에 따라 새로운 피처를 추가에 해당 값에만 1을 표시
    2. 이상치
        1. 표준점수(표준화)로 변환후 -3이하 +3 이하로 제거 (standardscaler)
        2. IQR - 분위수를 사용 (robust_scale)
        3. 정규분포
        4. 표준정규분포
        5. 6 - sigma
        6. 앤드류스 그림
        7. 마하라노비스 거리
    3. 피처 스케일링과 정규화
        1. 피처스케일링이란? 서로 다른 변수의 값 범위를 일정한 수준으로 맞추는 작업
        2. 표준화 - 평균이 0 이고 분산이 1인 정규 분포를 가진 값으로 변환/ 비교하기 용이하게 하기위해/ 확률 간편 계산 (StandardScaler)
        3. 정규화 - 서로 다른 피처의 크기를 통일하기 위해 변환 / 개별 데이터의 크기를 모두 똑같은 단위로 변경하는 것 (MinMaxScaler)
        4. StandardScaler - 개별 피처를 평균이 0 이고, 분산이 1인 값으로 변환(표준화)
        5. MinMaxScaler - 데이터값을 0과 1사이의 범위값으로 변환(정규화)
        6. RobustScaler - 1사분위와 3사분위로 하는 것(분류나 예측에 있어 산포를 더 크게 표준화 한다.)
        7. 데이터 분포 변환
            1. log - 로그 변환 (정규성을 높이고 분석에서 정확한 값을 얻기 위해 사용 / 데이터간의 편차를 줄일수 있다.+ 왜도(데이터가 치우친 정도) 첨도(뾰족한 분포)를 줄일수있다.)
            2. Sqrt - 제곱근 변환
        8. 데이터 단위 변환
            1. scalling - 평균이 0이고 분산이 1인 값으로 변환
            2. min-max scalling - 특정 범위로 모두 변환
            3. box-cox - 여러 k 값중 가장 작은 sse로 변환
            4. Robust_scale - median, interquartlie range 사용(아웃라이어 영향 최소화)
        9. 유의점
            1. 학습데이터와 테스트 데이터의 스케일링 변환시 유의점 처음에 fit했던걸로 transform 해야 한다. 아니면 서로 다르게 변환된다.
            전체 데이터의 스케일링 변환을 적용한 뒤 학습과 테스트 데이터 분리

4. 평가
    1. 정확도(Accuracy)
        1. 실제 데이터에서 예측 데이터가 얼마나 같은지를 판단하는 지표
        2. 직관적이긴 하지만 모델의 성능을 왜곡 할 수 있다.(불균형한 레이블 세트에서는 사용하면 안된다.)
        3. (TN + TP) / (TN + FP + FN + TP)
    2. 오차행렬
        1. TN - 부정이라고해서 맞은경우
        2. TP - 긍정이라고 해서 맞은경우
        3. FN - 부정이라고해서 틀린 경우
        4. FP - 긍정이라고 해서 틀린 경우
    3. 정밀도와 재현율
        1. 정밀도(Precision) - 긍정(양성 예측도)으로 예측한 것들로 비교 (TP / (FP + TP)) ex)스팸
        2. 재현율(Recall) - TP / (FN + TP) - 부정이라고 예측한 것들로 비교 ex)암
        3. 정밀도 / 재현율 트레이드 오프
            1. 임곗값 조정을 통해 한쪽의 수치를 높일 수 있다.
            2. 상호보완으로 써야 한다.(적절하게 써야함)
    4. F1 스코어
        1. 정밀도와 재현율을 결합한 지표(한쪽으로 치우치지 않을때 높은 값을 가진다.)
    5. ROC 곡선과 AUC - (이진 분류의 예측 성능 측정에서 중요하게 사용 되는 지표)
        1. ROC 곡선 - 수신자 판단 곡선, FPR(특이도)이 변할 때 TPR(민감도)이 어떻게 변하는 지를 타나내는 곡선
        2. 민감도(환자를 환자로)와 특이도(정상인을 정산인으로)
        3. TPR -1 / FPR -0(1-TPR)
        4. ROC 곡선이 1에서 가까워질수록 성능이 좋다.
        5. AUC는 면적
5. 분류
    1. 분류란
        1. 답(레이블)이 있는 데이터가 주어진 상태에서 학습하는 머신러닝 방식
    2. 결정트리
        1. 데이터에 있는 규칙을 학습을 통해 찾아내 트리 기반의 분류 규칙 생성(if/else)
        2. 많은 규칙 - 과적합으로 간다. 깊어질 수록 과적합일 가능성이 높다
        3. 엔트로피 - 데이터 집합의 혼잡도 / 다른 값이 섞여있으면 엔트로피가 높다.
        4. 특징
            1. 장점 - 쉽고 직관적 ,트리가 룰이 명확해 어떻게 구성되는지를 알수 있다.
            2. 단점 - 과적합으로 정확도가 떨어진다.
        5. 파라미터
            1. 노드를 분할하기 위한 최소 샘플 데이터수
            2. 노드가 되기위한 최소한의 샘플 데이터 수
            3. 최대 피처개수
            4. 트리 깊이
            5. 노드의 최대 개수
    3. 앙상블학습
        1. 여러 개의 분류기를 생성하고 그 예측을 결합함으로써 성능을 올리는 기법
        2. 보팅 - 여러개의 분류기가 투표를 통해 최종 예측 결과를 결정
            1. 하드 보팅 - 다수결 원칙(다수의 분류기가 결정한 예측값으로 선정)
            2. 소프트 보팅 - 분류기들의 레이블 값 결정 확률을 모두 더하고 이를 평균해서 이들 중 확률이 가장 높은 레이블 값을 최종값으로 선택

3. 배깅 - 각각의 분류기가 모두 같은 알고리즘이지만, 데이터 샘플링을 서로 다르게 하여 결과물 집계.(랜덤 포레스트)
4. 부스팅 - 여러개의 분류리가 순차적으로 학습을 수행하되, 틀린 데이터에 대해 가중치를 부여하면서 학습을 하는 방식
5. 스태킹 - 여러 가지 다른 모델의 예측 결괏값을 다시 학습데이터로 만들어 다른 모델로 재학습 시켜 결과를 예측하는 방법

4. 랜덤포레스트
    1. 쉽고 직관적이며 빠른 수행 속도
    2. 트리기반의 단점은 하이퍼 파라미터가 너무 많고 시간이 오래 걸린다.
5. GBM
    1. 약한 학습기를 순차적으로 학습 예측하면서 가중치를 부여해 학습하는 방식
    2. adaboost 비슷함
    3. gbm은 경사하강법을 쓴다.
6. XGBoost
    1. gbm 기반
    2. 장점
        1. 빠른 수행 시간
        2. TreePruning(이득이 없는 분할을 가지치기 할 수 있다.) - 과적합 규제
        3. 자체 내장된 교차 검증 / 성능평가 / 피처 중요도
        4. 결손값 처리
        5. 병렬 CPU 환경에서 병렬 학습 가능
    3. c/c++로 구성
    4. 파이썬 래퍼 모듈 / 사이킷런 래퍼 모듈이 있음
7. LightGBM
    1. 장점
        1. 메모리 사용량이 더 적다
        2. XGB보다 빠르다.
        3. 카테고리형 피처의 자동 변환과 최적 분할
    2. 단점- 적은 데이터에서 과적합에 취약
    3. 리프 중심 트리 분할 방식 사용 leaf wise() - 최대 손실 값을 가지는 리프노드를 분할 / 일반적으로 균형 트리 분할(Level Wise) - 오버피팅에 강함 but 시간이 오래걸림
8. 스태킹 앙상블
    1. 개별 결과 데이터 세트를 최종적인 메타 데이터 세트로 만든다.

6. 회귀
    1. 회귀란
        1. 예측 값이 연속형 숫자값 / 일반 선형 회귀 - 예측값과 실제값의 차이인 RSS를 최소화 할 수 있도록 회귀 계수를 최적화하며 규제를 안하는 모델
    2. 단순 선형회귀
        1. 독립변수 하나 종속 변수 하나
        2. 최적의 회귀 계수 -> 전체 데이터의 잔차(예측값과 실제값의 차이) 합이 최소가 되는 모델을 만든다.
        3. 보통 절대값(MAE)이나 제곱(RSS)을 구해 더하는 방식을 사용
    3. 비용최소화(경사하강법)
        1. 점진적인 방법으로 오류 값이 최소가 되는 W 파라미터를 구하는 방식
    4. 다항회귀와 과대적합과 과소 적합
        1. 다항 회귀란 독립변수가 2차, 3차 방정식 같은 다항식으로 표현되는 것
        2. 편향-분산트레이드 오프 - 머신러닝이 극복해야 할 가장 중요한 이슈
        3. 과대적합 - 분산이 높으면 편향이 낮아진다
        4. 과소적합 - 편향이 높으면 분산은 낮아진다.
        5. 편향과 분산이 서로 트레이드오프를 이루면서 오류 cost값이 최대로 낮아지는 모델을 구축하는 것이 가장 효율적인 머신러닝 예측 모델이다.
    5. 규제 선형모델 - 릿지,라쏘, 엘라스틱넷 , 서포트 벡터 머신

1. 라쏘 - L1 규제를 적용 (상대적으로 작은 회귀 계수의 값을 0으로 만드는 방법 - 필요 없는 피처를 빼기위해예측 영향도를 줄이기 위해)
2. 릿지 - L2 규제를 적용 (상대적으로 큰 회귀 계수의 값을 감소 시키는 방법 - 예측 영향도를 줄이기 위해)
3. 엘라스틱넷 - L2 와 L1을 결합한 모델

6. 선형 서포트 벡터 머신 - 선을 그어서 분류를 하는 알고리즘(선을 긋는 함수는 최대 마진 초평면이라고 한다. 초평면을 정의할때 사용되는 점을 서포트 벡터라고 한다.)
7. 서포트 벡터 머신 - 커널함수라로 불리는 비선형적으로 2개의 벡터거리를 측정하는 함수로 치환한다.
8. 선형회귀 모델을 위한 데이터 변환
    1. StandardScaler(표준화) - 평균이 0 분산이 1인 정규 분포를 가진 데이터 세트로 변환 / MinMaxScaler(정규화) - 최솟값이 0 이고 최대값이 1인 값으로 정규화
    2. 다항 특성을 적용하여 변환하는 방법
    3. 로그 변환 - log 함수를 적용해야 정규 분포에 가까운 형태로 값이 분포
9. 로지스틱 회귀
    1. 선형 회귀 방식을 분류에 적용한 알고리즘
    2. 분류에 사용 된다.
    3. 시그모이드 함수 최적선을 찾고 이 시그모이드 함수의 반환 값을 확률로 간주해 확률에 따라 분류를 결정하는 것
10. 회귀트리
    1. 리프 노드에 속한 데이터 값의 평균값을 구해 회귀 예측값을 계산
11. 회귀 평가 지표
    1. MAE - 실제 값과 예측값의 차이를 절댓값으로 변환해 평균
    2. MSE - 실제값과 예측값의 차이를 제곱해 평균한 것
    3. RMSE - MSE에 루트를 씌운것(오류 평균이 커지는 특성 때문에 루트 사용)
    4. R제곱 - 분산 기반으로 예측 성능을 평가 (1이 게까울수록 정확도가 높다)

7. 차원축소
    1. 차원축소란 - 많은 피처로 구성된 다차원 데이터 세트의 차원을 축소해 새로운 차원의 데이터 세트를 생성하는 것
        1. 피처 선택 - 데이터의 특징을 잘 나타내는 피처만 선택
        2. 피처 추출 - 기존 피처를 저차원의 중요 피처로 압축해서 추출하는 것
    2. 장점
        1. 잠재적인 요소 추출
        2. 과적합 영향력이 적어진다
        3. 텍스트 문서의 숨겨진 의미 추출
    3. 주성분 분석(PCA)
        1. 여러 변수간에 존재하는 상관(수치적)관계를 이용해 이를 대표하는 주성분을 추출해 차원 축소
        2. 변동성이 가장 큰 축을 찾는 방법
        3. 목표변수를 고려하여 목표변수를 잘 예측할 수 있는 선형결합으로 이루어져 있는 몇개의 주성분을 찾아낸다.
        4. 선형적 결합이 중심
        5. 데이터 세트를 저차원 공간에 투영해 차원을 축소
        6. 데이터 유실이 최소화
        7. 가장 높은 분산을 가지는 데이터의 축을 찾아 이 축으로 차원 축소 - 주성분
        8. 가장 큰 데이터 변동성을 기반으로 첫번째 벡터 축을 생성 , 두번째는 이에 직각이 되는 벡터를 축으로
        9. 순서
            1. 입력 데이터 세트의 공분산 행렬을 생성
            2. 공분산 행렬의 고유벡터와 고유값을 계산
            3. 고유값이 가장 큰순으로 K개만큼 고유벡터를 추출

4. 고유값이 가장 큰 순으로 추출된 고유벡터를 이용해 새롭게 입력 데이터를 변환

4. 요인분석(EFA)
    1. 잠재구조 결합이 중심
    2. 개념적 논리적으로 비슷한 변수들을 잠재적인 요소로 결합한다
    3. 목표 변수를 고려하지 않고 데이터들간의 상관성을 토대로 비슷한 변수들을 묶어 사용
5. LDA (선형 판별 분석법)
    1. 입력 데이터의 결정 값 클래스를 최대한으로 분리할 수 있는 축을 찾는 방식
    2. 지도 학습의 분류에서 사용하기 쉽도록 개별 클래스를 분별할 수 있는 기준을 최대한 유지하면서 차원 축소를 한다.
    3. 클래스 간 분산은 최대한 크게 가져가고 클래스 내부의 분산은 최대한 작게 가져가는 방식
    4. LDA와 PCA의 차이는 PCA는 공분산행렬을 사용하고 LDA는 클래스간 분산과 클래스 내부 분산행렬을 생성하여 사용
    5. 순서
        1. 클래스 내부와 클래스간 분산 행렬 구한뒤 입력데이터의 결정 값 클래스별로 개별 피처의 평균 벡터를 기반으로 구한다
        2. 클래스 내부와 클래스간 분산 행렬 ->고유벡터로 분해
        3. 고유값이 가장 큰 순으로 K개 차출
        4. 고유값이 가장 큰 순으로 추출된 고유벡터를 이용해 새롭게 입력 데이터를 변환
6. SVD
    1. 고차원 행렬을 두 개의 저차원 행렬로 분리하는 행렬 분해 기법
    2. 정방행렬뿐만아니라 크기가 다른 행렬에도 적용할 수 있다.
    3. 특이값 분해
7. NMF
    1. 낮은 랭크를 통한 행렬근사 방식
    2. 양수 행렬로 분해될 수 있는 기법

8. 군집화
    1. 군집화란 ? - 개체들이 주어졌을 때, 몇개의 클러스터로 나누는 작업
    2. K-평균알고리즘(k-means clustering)
        1. 군집 중심점이라는 특정한 임의의 지점을 선택해 해당 중심에 가장 가까운 포인트들을 선택하는 군집화 기법
        2. 평균이 중심에 소속된 데이터의 평균 거리 중심으로 이동
        3. 원형의 범위로 구한다.
        4. 타원일 경우 제대로 구하지 못함
    3. 계층적 클러스터링 - 근접한 데이터를 결합함으로써 클러스터링하는 방법
    4. 평균 이동(mean shift)
        1. 중심을 데이터가 모여 있는 밀도가 가장 높은곳으로 이동
        2. 데이터의 분포도를 이용해 군집 중심점을 찾는다.
        3. 확률밀도 함수 사용
        4. KDE 사용
    5. GMM(Gaussian Mixture Model)
        1. 가우시안 분포를 가진 데이터 집합들이 섞여서 생성된 것이라는 가정하에 군집화 수행 방식
        2. 여러개의 정규 분포 곡선을 추출한뒤, 개별 데이터가 이 중 어떤 정규 분포에 속하는지 결정하는 방식
        3. 타원형에서도 잘 표현 된다.
        4. 수행시간이 오래걸린다.
    6. DBSCAN(Density Based Spatial Clustering of Applications with Noise)
        1. 밀도 기반 군집화
        2. 특정 공간내에 데이터 밀도 차이를 기반 알고리즘
        3. 데이터의 분포가 기하학적으로 복잡한 데이터 세트에도 효과적인 군집화가 가능
        4. 입실론 - 개별 데이터를 중심으로 입실론 반경을 가지는 원형의 영역

5. 최소 데이터 개수 - 개별 데이터의 입실론 주변 영역에 포함되는 타 데이터의 개수
6. 즉 입실론 영역 내에 포함되는 최소 데이터 개수를 충족시키는가 아닌가에 따라 데이터 포인트를 정의한다

7. 군집평가
    1. 실루엣 계수가 1에 가까워야함
    2. 개별 군집의 평균값의 편차가 크지 않아야 한다.
    3. 실루엣 분석 - 군집화 평가 방법으로서 각 군집 간의 거리가 얼마나 효율적으로 분리돼 있는지 나타내는 것
        1. 실루엣 계수 - 개별 데이터가 가지는 군집화 지표
        2. 1에가까워질수록 근처의 군집이 멀리있다는것 0 에 가까울수록 근처의 군집과 가까워진다는 것
8. 고객 세그먼테이션(군집화) - 타깃 마게팅
    1. 고객을 여러 특성에 맞게 세분화해서 그 유형에 따라 맞춤형 마케팅이나 서비스를 제공하는 것
    2. RFM 기법 - 가장 최근 상품 구입 일 에서 오늘 까지의 기간(Recency), 상품 구매 횟수(Frequency), 총 구매 금액(Monetary Value)

9. 텍스트분석
    1. 텍스트 분석이란
        1. NLP - 머신이 인간의 언어를 이해하고 해석하는데 중점
        2. 텍스트 분석 - 비정형 텍스트에서 의미 있는 정보를 추출하는 것에 중점
    2. 텍스트 분석 수행 프로세스
        1. 텍스트 사전 준비 작업
        2. 피처 벡터화 추출 - 피처를 추출하고 여기에 벡터 값을 할당, 대표적인 방법으로 BOW , Word2Vec이 있으며, BOW 는 대표적으로 Count 기반과 TF-IDF 기반 벡터화가 있다.
        3. ML 모델 수립 및 학습/예측/평가
        4. Word2Vec - cbow (주변단어로 중심단어를 예측), skip-gram (중심단어로 주변 단어를 예측)
    3. 분석 패키지
        1. NLTK
        2. gensim
        3. SpaCy
    4. 텍스트 사전준비작업
        1. 클렌징 - 불필요한 문자 제거
        2. 텍스트 토큰화
            1. 문장 토큰화 - 문서에서 문장을 분리하는 문장 토큰화(. 이나 개행문자등 마지막을 뜻하는 기호에 따라 분리하는 것이 일반적 , 정규표현식도 가능)
                1. 시맨틱적인 의미가 중요한 요소로 사용될 때 많이 사용
            2. 단어 토큰화 - 단어를 토큰으로 분리(공백 콤마, 마침표, 개행문자, 정규표현식)
        3. 스톱워드 제거 - 분석에 큰 의미가 없는 단어(스톱워드 목록화를 통해 삭제 ex)is a , he, him 등)
        4. Stemming과 Lemmatization -문법적 또는 의미적으로 변화하는 단어의 원형(어근)을 찾는것
            1. Stemming - 단어 자체만 고려
            2. Lemmatization - 품사와 같은 문법적인 요소를 감안해서 어근을 찾는다
    5. BOW()
        1. 문서가 가지는 모든 단어를 문맥이나 순서를 무시하고 일괄적으로 단어에 대해 빈도값을 부여해 피처 값을 추출하는 모델
        2. 단점
            1. 문맥의미 반영 부족
            2. 희소 행렬 문제
        3. 피처 벡터화
            1. 카운트 기반 벡터화 - 카운트 값이 높을수록 중요한 단어 (CountVectorizer)

2. TF-IDF - 카운트 기반이긴 하되, 모든 문서에서 전반적으로 자주 나타나는 단어에 대해서는 패널티를 주는 방식(TfidVectorizer)
4. 희소행렬 처리 방법
1. coo - 0이 아닌 데이터만 별도의 데이터 배열에 저장하고, 그 데이터가 가리키는 행과 열의 위치를 별도의 배열로 저장하는 방식(row와 칼럽을 별도의 배열로 저장)
1. ex) [3,1,2] -> (0,0)(0,2)(1,1) -> [0,0,1] [0,2,1]
2. csr - 행 위치 배열내에 있는 고유한 값의 시작 위치만 다시 별도의 위치 배열로 가지는 변환 방식
1. ex)[0,0,1,1,1,1,1,2,2,3,4,4,5] -> [0,2,7,9,10,12,13] 마지막 값에는 총 항목 개수를 배열에 추가
6. 사이킷런 파이프 라인 사용 및 Grid SearchCV와의 결합
1. 텍스트 기반의 벡터화 뿐만아니라 모든 데이터 전처리 작업과 Estimator를 결합
2. ex) 스케일링, 벡터 정규화, PCA등의 변환 작업과 분류 회귀등의 Estimator를 한 번에 결합
7. 감성분석
1. 문서의 주관적인 감성/의견/감정/기분 등을 파악하기 위한 방법
2. ex) 소셜미디어 , 여론조사, 온리안 리뷰, 피드백
3. 지도학습 - 학습데이터와 타깃 레이블 값을 기반으로 감성 분석 학습을 수행한뒤 이를 기반으로 다른 데이터 예측
4. 비지도학습- 감성 어휘 사전(Lexicon)을 통해 사용
5. 시맨틱 - 문맥상 의미
6. sentiWordNet - NLTK 패키지의 WordNet과 유사하게 감성단어 전용 WordNet 구현, 3가지 감성점수를 할당 긍정 감성, 부정 감성, 객관성 지수
7. VADER - 소셜 미디어의 텍스트에 대한 감성 분석을 제공하기 위한 패키지
8. 토픽 모델링
1. 문서 집합에 숨어있는 주제를 찾아내는 것
2. LSA(Latent Semantic Analysis - 잠재 의미 분석) - 차원축소(svd)를 사용하여 문서에 숨어있는 의미를 찾는다
3. LDA(Latent Dirchlet Allocation - 잠재 디레클레 할당) - 단어가 특정 토픽에 존재할 확률과 문서에 특정 토픽이 존재할 확률을 결합 확률로 토픽을 추출
9. 문서 군집화 소개와 실습
1. 비슷한 텍스트 구성의 문서를 군집화
2. 비지도 학습
3. 군집별 핵심 단어 추출 가능
10. 문서 유사도
1. 코사인 유사도를 사용
2. 벡터의 크기 보다는 벡터의 상호 방향성일 얼마나 유사한지에 기반(벡터 사이의 사잇각을 구한다.)
11. 한글 텍스트 처리
1. 조사 때문에 어려움
2. 어근 추출 힘듬
3. 띄어쓰기 문제
4. KoNLPy 소개
12. 텍스트 분석
1. 데이터 전처리
2. 피처 인코딩
3. 피처 벡터화
4. 모델 사용
10. 추천시스템
1. 추천시스템의 개요와 배경
1. 사이트를 평가하는 중요한 요소가 될 정도
2. 추천시스템의 유형
1. 콘텐츠 기반 필터링(CBF)
2. 협업 필터링(CF)
1. 최근접 이웃 협업 필터링

1. 사용자 기반 협업 필터링
2. 아이템 기반 협업 필터링
2. 잠재요인 협업 필터링 - 이게 많이 뜬다
3. 하이브리드 형식(콘텐츠 기반과 협업 기반을 적절히 결합해 사용)

2. 콘텐츠 기반 필터링 추천시스템
    1. 특정한 아이템을 선호하는 경우 그 아이템과 비슷한 콘텐츠를 가진 다른 아이템을 추천하는 방식
3. 최근접 이웃 협업 필터링
    1. 사용자 행동 양식만을 기반으로 추천을 수행
    2. 목표 - 축적된 사용자 행동 기반으로 사용자가 평가하지 않은 아이템을 예측 평가하는 것
    3. 사용자 기반 - 당신과 비슷한 고객들이 다음 상품도 구매했다 (A사용자 B사용자 가 나올때 두 사용자가 유사하니 A사용자가 좋게 봤던걸 B한테 추천) 많은 사람들이 소비한것은 정보력이 떨어진다.
    4. 아이템 기반 - 이 상품을 선택한 다른 고객들이 다음 상품도 구매했다(A아이템과 B 아이템에 대한 사용자들의 평점이 비슷할 경우 B 아이템을 추천) 이것이 효과가 더 좋다. 대부분 이것을 사용
    5. TOP-N으로 선정해 사용자가 좋아하는 아이템을 추천
4. 잠재 요인 협업 필터링
    1. 사용자 - 아이템 평점 매트릭스속에 숨어 있는 잠재 요인을 추출해 추천 예측을 하는 기법
    2. 행렬 분해 기반
    3. SVD는 널값이 없는 행렬에만 적용할 수 있다. 그래서 이런 경우에는 SGD / ALS 방식을 이용해 SVD를 수행한다.
        1. SGD - 확률적 경사 하강법을 이용한 행렬분해
            1. P와 Q를 임의의 값을 가진 행렬로 설정
            2. P와 Q.T값을 곱해 예측 R행렬을 계산하고 예측 R행렬과 실제 R행렬에 해당하는 오류 값을 계산
            3. 오류 값을 최소화할 수 있도록 P와 Q행렬을 적절한 값으로 각각 업데이트
            4. 만족할 만한 오류값을 가질 때 까지 2,3 번 작업을 반복하면서 P와 Q값을 업데이트해 근사화 한다.
        2. ALS - 사용자와 아이템의 잠재요소를 번갈아 가며 학습하는 방식
            1. 사용자 혹은 아이템의 Latent Factor 행렬을 아주 작은 랜덤값으로 초기화
            2. 둘 중 하나를 상수처럼 고정시켜 Loss Function 을 Convex Function으로 만든다
            3. 이를 미분한 다음, 미분 값을 0으로 만드는 사용자 혹은 아이템의 Latent Factor 행렬을 계산
            4. 이 과정을 사용자 한번, 아이템 한번 반복하면서 최적의 x,y를 찾아낸다.
5. 콘텐츠 기반 필터링
6. 아이템 기반 최근접 이웃 협업 필터링
7. 행렬분해를 이용한 잠재요인 협업 필터링
8. 파이썬 추천 시스템 패키지 Surprise

11. 분석 방식
    1. 데이터 클렌징 및 가공
    2. 모델링 사용
    3. 로그변환
    4. 피처 인코딩
    5. 모델학습 예측 평가
12. 오버피팅- 학습 데이터에만 최적화 되는 경우 /공부
    1. 교차검증
    2. 적극적인 정규화
    3. 모델을 더 간단하게 만드는 방법(변수 줄이기)
    4. 모델을 덜 민감하게 만드는 법(구분선을 구성하는 파라메터 또는 계수의 민감도 줄이기)

    5. 더 많은 데이터 학습
13. 언더피팅 - 데이터가 적어 모델이 학습이 제대로 되지 않는 경우 /공부
14. 결측치의 종류
    1. 완전 무작위 결측 - 다른 변수들과 아무런 상관이 없는 경우
    2. 무작위 결측 - 값 자체의 상관관계는 알 수 없는 경우
    3. 비무작위 결측 - 누락된 값이 다른 변수와 연관이 있는 경우
15. 결측치 가이드라인
    1. 10프로 미만 - 삭제
    2. 10~20 - hot deck
    3. 20~50 - regression or model based imputation
    4. 50 - 삭제
    5. 근데 정해진건 아님
16. single Imputation
    1. hot-deck - 최빈값
    2. mean imputation - 평균값
    3. regression imputation - 회귀추정해서 넣자
17. multi Imputation - 여러개의 데이터셋을 만들어 평가
18. 대처법의 종류
    1. 제거법(완전 제거법/ 한쌍 제거법) - 결측치가 있는 행자체를 지워버리는 방법 / 결측을 포함하는 응답자를분석에서 제외하고, 남아 있는 관측치에 대해 통계분석을 시행 #제거법의 단점 - 표본의 수가 줄어들어 통계적 검정력이 떨어짐
    2. 단일 대체법
        1. 평균 대체 - 평균으로 대체
        2. 최빈값, 중앙값, 0으로 처리, na 로 처리
        3. 회귀 대체 - 설명변수의 조건부 평균으로 결측을 대체
        4. 단점 - 편의 추정량을 발생, 통계적 검정력의 관점에서도 문제
    3. 다중 대체법 - 결측치를 제외한 나머지 변수들로 해당 결측치를 예측