Project #2

DB Mining & Recommender System & Document Search Engine & Classification

과제목적 : 이번 프로젝트는 프로젝트 1 에서의 결과물을 바탕으로 한 DB 마이닝 및 추천 시스템 구현, 텍스트 데이터를 바탕으로 한 검색 엔진 모듈 및 분류 모델 구현을 목적으로 한다.

프로젝트는 크게 네 부분으로 나뉜다.

PART Ⅰ. 연관 분석

PART Ⅱ. 추천 시스템

PART Ⅲ. 문서 검색 엔진

PART Ⅳ. 문서 분류

PART Ⅰ. 연관 분석

PART Ⅰ 연관분석에서는 사이트 A 의 포스트 간의 연관 분석을 목표로 한다. R1-1 에서는 연관분석을 위한 horizontal table 의 결과를 반환하는 것을 목표로 하며, R1-2 에서는 python 의 mlxtend 라이브러리를 사용하여 연관분석을 수행하고 결과를 출력하는 것을 목표로 한다.

(R1-1) (10%) 결과물: DMA_project2_team##_part1_horizontal.pkl

연관분석을 위해 pandas로 DMA_project_UPI.csv를 Load하라. 이후, user id를 index로 가져야 하며, post id들을 column 명으로 가지는 DataFrame을 만들어라. DMA_project_UPI.csv에 해당 user의 해당 post에 대한 IntDegree 정보가 있다면 1, 없다면 0을 저장해야 한다. 저장된 DataFrame의 각 user는 연관분석의 transaction 역할을, 각 post는 연관분석의 item 역할을 하게 된다. Pandas 라이브러리를 사용하여 해당 horizontal table을 생성하고, 이를 DMA_project2_team##_part1_horizontal.pkl에 저장하라.

(R1-2) (10%) 결과물: DMA_project2_team##_part1_association.pkl

R1-1에서 만든 DataFrame을 사용해서 다음의 조건을 만족하는 frequent itemset을 만들고 연관분석을 수행하라. 그리고 이 결과에 대한 간략한 정성적, 정량적 평가를 수행하라.

- Frequent itemset의 최소 support: 0.2
- 연관분석 metric: lift (lift >= 2 인 것들을 출력)
- 결과 파일명: DMA_project2_team##_part1_association.pkl

PART Ⅱ. 추천 시스템

PART Ⅱ 에서는 사용자들에게 포스트를 추천하는 추천시스템 구현을 목적으로 한다. 추천시스템 구현시 TODO 파일을 활용해야 한다. R2-1에서는 점수 예측 결과를 반환하는 함수 작성을, R2-2부터 R2-4까지는 recommendation system을 구현하도록 한다. 이 때 라이브러리로는 surprise를 이용한다.,

(R2-1) (3%)

점수 예측 결과 top-n개 결과를 반환하는 get_top_n 함수를 작성하라. 세부 내용은 뼈대 코드를 참고하여 작성하도록 한다.

(R2-2) (6%) User-based Recommendation 결과물: 2-2-1.txt, 2-2-2.txt

['1496', '2061', '2324', '4041', '4706'] 의 총 5명 user에 대하여 다음의 알고리즘과 유사도 함수를 사용한 추천 결과 top-TODO post를 텍스트 파일로 출력하라.

- 알고리즘 : KNNBasic 유사도: cosine 파일명: 2-2-1.txt
- 알고리즘 : KNNWithMeans 유사도: pearson 파일명: 2-2-2.txt

또한, User-based recommendation에서 다양한 알고리즘과 유사도 함수를 적용해보고 cross validation(k=TODO, random_state=TODO)을 기준으로 가장 좋은 성능을 보이는 모델을 제출하라.

(R2-3) (6%) Item-based Recommendation 결과물: 2-3-1.txt, 2-3-2.txt

['20', '45', '48', '139', '162'] 의 총 5개 post에 대해 다음 알고리즘과 유사도 함수를 사용한 추천 결과 top-TODO user를 텍스트 파일로 출력하라.

- 알고리즘 : KNNBasic 유사도: cosine 파일명: 2-3-1.txt
- 알고리즘 : KNNWithMeans 유사도: pearson 파일명: 2-3-2.txt

또한, Item-based recommendation에서 다양한 알고리즘과 유사도 함수를 적용해보고 cross

validation(k=TODO, random_state=TODO)을 기준으로 가장 좋은 성능을 보이는 모델을 제출하라.

(R2-4) (10%) Matrix-based Recommendation 결과물: 2-4-1.txt, 2-4-2.txt, 2-4-3.txt, 2-4-4.txt

['1496', '2061', '2324', '4041', '4706'] 의 총 5명 user 에 대하여 다음의 알고리즘을 사용한 추천 결과 top-TODO post을 텍스트 파일로 출력하라.

- SVD(n_factors=100, n_epoch=50, biased=False) 파일명: 2-4-1.txt
- SVD(n_factors=200, n_epoch=100, biased=True) 파일명: 2-4-2.txt
- SVD++(n_factors=100, n_epoch=50) 파일명: 2-4-3.txt
- SVD++(n_factors=100, n_epoch=100) 파일명: 2-4-4.txt

또한, Matrix-based recommendation에서 다양한 알고리즘과 유사도 함수를 적용해보고 cross validation(k=TODO, random_state=TODO)을 기준으로 가장 좋은 성능을 보이는 모델을 제출하라.

PART Ⅲ. 문서 검색 엔진

PART Ⅲ에서는 주어진 질의어와 관련이 높은 순서대로 문서들을 나열하는 검색 엔진 모듈을 python의 whoosh 라이브러리를 사용하여 구현하여야 한다.

- 사용 데이터: LISA 데이터셋
  - document.txt: 6004개 문서 파일
  - query.txt: 35개 질의어 파일
  - relevance.txt: 각 질의어의 실제 연관 문서가 명시된 정답 파일

- 작성 모듈
  - [선택] make_index.py: 문서의 ID와 contents를 index에 저장하는 함수. 기본으로 주어진 index를 사용하지 않을 시에 해당 모듈 작성
  - QueryResult.py: 텍스트 형태의 질의어를 입력 받아 whoosh 질의어 객체로 변환 후 검색 결과 반환
  - CustomScoring.py: 문서들을 질의어와 관련 높은 순서대로 나열할 때 사용하는 문서 채점 함수. 사용 가능한 기본 정보로는

    ✓ 문서 내 단어 빈도 (TF)
    ✓ 역문서 빈도 (IDF)
    ✓ 전체 데이터셋 내 단어 빈도
    ✓ 문서 개수
    ✓ 문서 길이 (단어 개수)
    ✓ 전체 데이터셋 내 단어 개수
    ✓ 문서 당 평균 단어 개수

    등이 있으며, 제공되는 정보 이외의 정보를 추출 가능하다면 추가적으로 사용 가능

- 평가 방법
  - 35개의 질의어 중 임의로 정해진 15개의 test 질의어에 대한 검색 성능 평가
  - 평가 지표로는 BPREF 사용 [evaluate.py에서 자동으로 계산]

$$\text{BPREF} = \frac{1}{R}\sum_{r}\left(1 - \frac{|n\ ranked\ higher\ than\ r|}{N}\right)$$

    - R : 연관 문서의 수
    - N : 비연관 문서의 수
  - **점수 산정: 총 25점**
    - ✓ 성능 평가 점수 15점: 15개 test 질의어 각각의 BPREF에 대해 상대평가 적용, [0.5, 1] scale하여 합산
    - ✓ 성능 개선 방법에 대한 근거와 정당화 내용 10점: 절대평가
      - 참고문헌 작성 여부
      - 제시한 근거와 결과가 논리적으로 타당할 경우

- 주의 사항
  - 제공되는 질의어에 test 질의어가 포함된 프로젝트 상황 특성을 활용한 **질의어–문서 매칭 금지**

    => 성능 개선 방법에 대한 근거 점수, 성능 평가 점수 모두 0점
  - 문서 분석은 허용, 질의어 분석은 금지
  - 결과 파일이 보고서에 명시한 내용과 실행 결과가 다를 경우 불이익 있음

PART IV. 문서 분류

PART IV 에서는 6 가지 카테고리의 영어 신문 기사를 분류하는 모델을 python 의 sklearn 라이브러리를 사용하여 구현한다.

- crime, entertainment, politics, science, space, technology 총 6개의 카테고리
- 각 카테고리마다 약 300~400개의 기사 데이터 제공
- text 폴더 내부에 train/test 폴더 구분, 각 폴더 내에 6개의 카테고리를 이름으로 하는 폴더 내에 텍스트 파일로 존재
- test 폴더에는 각 카테고리별 기사 개수의 20%(약 50~90개 정도)의 텍스트 파일 존재
- 주어진 데이터 외에 추가로 크롤링 등의 방법을 이용한 **데이터 추가, 파일 수정, 제공된 데이터셋 일부만 사용은 허용하지 않음**
- 2024.12.9(프로젝트 마감일) 이후 작성된 기사 20개에 대해 올바르게 분류한 개수로 성능 평가
- 각 모델에 대해 학습시킨 모델을 pickle 파일로 저장하여 코드와 함께 제출

- 평가 방법
  - 4-1. Naïve Bayes Classifier
    - ✓ 점수 10점: 20개 신문 기사 중 올바르게 분류한 개수 * 0.5
  - 4-2. SVM
    - ✓ 점수 10점: 20개 신문 기사 중 올바르게 분류한 개수 * 0.5

- **채점 기준**

- **PART Ⅰ(20%):** Horizontal (10%), Association (10%)
- **PART Ⅱ(25%):** get_top_n (3%), 각 추천 결과 파일 (2%), 각 cv 기준 best model (2%)
- **PART Ⅲ(25%):** 성능평가(15%), 성능 개선 방법에 대한 근거와 정당화 내용(10%)
- **PART Ⅳ(20%)**: 분류 모델 성능평가(10%, 10%)
- **보고서 품질(5%), 발표(5%)**

- **강의계획서에 기재된 바와 같이 프로젝트의 성적 반영 비중은 두 번의 프로젝트를 합쳐 30%이다.**

결과물들을 'DMA_project2_team##.zip'파일로 압축하여 발표일 전날인 12월 8일 23:59까지 ETL에 업로드해야 한다. ETL 상에 문제가 생겼을 경우 mo970610@snu.ac.kr 로 오류 증명 파일과 함께 해당 일시까지 보내야 한다. 이 외 문제가 발생할 경우 사전 공지사항 및 제출해야 할 결과물과 파일명, 파일 확장자는 다음과 같다.

- 보고서 (20페이지 이내)
  - 파일명: DMA_project2_team##_보고서.pdf
- 발표 자료 및 발표
  - 발표 자료 파일명: DMA_project2_team##_발표자료.pdf
- Python 프로그램 코드 및 결과물
  - 사용 라이브러리로는 pandas, numpy, sklearn 및
    (PART Ⅰ, Ⅱ) collections, mlxtend, surprise
    (PART Ⅲ, Ⅳ) nltk, whoosh, pickle을 기반으로 한다.
    필요하다면 다른 라이브러리를 사용할 수 있으나 이에 대해 (PART Ⅰ, Ⅱ의 경우) alswo5131@snu.ac.kr 또는 (PART Ⅲ, Ⅳ의 경우) mo970610@snu.ac.kr 로 사전에 메일을 보내야 한다.
  - 문서 검색 엔진 구현에서 사용하려는 방법이 허용되는지 **애매할 경우 이메일 문의**
  - csv 파일은 column들 사이에 분리 기호는 콤마(,)여야 하며 row들 사이에는 줄 넘김(₩n)으로 구분되어야 한다. Query 실행 결과 또는 view에 대한 csv를 저장할 땐 workbench 상에서 저장하는 것이 아니라 python 코드를 통해 저장을 수행하여야 한다.

- Pandas의 DataFrame을 pkl으로 저장할 때는 to_pickle 함수를 이용하여 저장한다.
- AA 폴더 (PART Ⅰ)
  - ✓ DMA_project2_team##_part1_horizontal.pkl
  - ✓ DMA_project2_team##_part1_association.pkl
- RS 폴더 (PART Ⅱ)
  - ✓ 2-2-1.txt
  - ✓ 2-2-2.txt
  - ✓ 2-3-1.txt
  - ✓ 2-3-2.txt
  - ✓ 2-4-1.txt
  - ✓ 2-4-2.txt
  - ✓ 2-4-3.txt
  - ✓ 2-4-4.txt
- SE 폴더 (PART Ⅲ)
  - ✓ index 폴더: 주어진 index 혹은 신규 생성한 index
  - ✓ make_index.py: 주어진 index를 사용하지 않았을 경우 첨부
  - ✓ CustomScoring.py
  - ✓ QueryResult.py
- CL 폴더 (PART Ⅳ)
  - ✓ classification.py
  - ✓ DMA_project2_team##_nb.pkl: Naïve Bayes Classifier 이용한 모델
  - ✓ DMA_project2_team##_svm.pkl: SVM 이용한 모델
- PART Ⅲ, Ⅳ 의 경우 추가 데이터 및 데이터 파일 가공은 허용하지 않았기에 첨부할 필요 없음.
- **각 모델의 결과를 제출 코드만을 실행하여 재현할 수 있어야 한다.**