v.1

Apriori 알고리즘을 사용하여 confidence가 0.7 이상인 순서쌍을 찾기 위해 Python 코드를 작성해 보겠습니다. 주어진 코드를 기반으로 데이터를 MySQL에서 불러와서 처리하고, Apriori 알고리즘을 적용하여 원하는 결과를 추출할 것입니다.

1. 데이터베이스 연결 및 데이터 불러오기

기존 코드에서 데이터를 MySQL 데이터베이스에 저장한 후, 필요한 데이터를 불러오는 방식을 사용하겠습니다. 데이터베이스 연결 및 데이터 불러오기 코드는 다음과 같습니다:

2. Apriori 알고리즘 구현

이제 Apriori 알고리즘을 직접 구현하여 confidence가 0.7 이상인 순서쌍을 찾겠습니다. 여기서는 itertools를 사용하여 가능한 모든 조합을 생성하고, 이를 통해 support와 confidence를 계산할 것입니다.

설명

데이터는 MySQL에서 불러와서 각 transaction을 생성하고, reconnaissance, infection, action을 제외한 나머지 요소들을 'on' 또는 'off'로 변환하여 transactions 리스트에 저장합니다.

Apriori 알고리즘은 후보 생성과 지지도 계산을 반복하여 frequent itemset을 생성합니다.

각 frequent itemset에 대해 가능한 규칙을 생성하고, 이 규칙의 confidence를 계산하여 주어진 min\_confidence 이상인 규칙을 출력합니다.

이 코드는 주어진 데이터에서 Apriori 알고리즘을 구현하고, confidence가 0.7 이상인 연관 규칙을 찾아 출력하는 기능을 수행합니다.

v.2

코드 설명

MySQL 데이터 삽입 부분: output-dataset\_ESSlab.csv 파일을 읽어 MySQL 데이터베이스에 삽입하는 부분입니다. INSERT INTO 쿼리를 사용하여 각 행을 데이터베이스에 삽입합니다.

데이터 전처리 및 Apriori 알고리즘 적용: MySQL에서 데이터를 가져와서 Apriori 알고리즘을 적용하는 부분입니다. 데이터를 transactions 리스트로 변환하고, 이를 기반으로 지지도 계산 및 Apriori 알고리즘을 적용하여 연관 규칙을 추출합니다.

결과 출력: 추출된 연관 규칙을 출력합니다.

위 코드를 실행하기 전에 MySQL 데이터베이스가 실행 중이고, 정상적으로 접속할 수 있는지 확인해야 합니다. 또한 코드에서 사용하는 계정 정보나 포트 번호가 MySQL 설정과 일치하는지도 확인해야 합니다.

Mysql

확장자가 .sql인 파일은 일반적으로 SQL 쿼리와 명령어를 포함하고 있으며, MySQL 데이터베이스에 대해 실행해야 하는 경우가 많습니다. SQL 파일을 데이터베이스에 적용하려면 다음과 같은 방법을 사용하여 파일의 내용을 실행할 수 있습니다.

방법 1: MySQL 클라이언트에서 실행

MySQL 클라이언트를 사용하여 SQL 파일을 실행할 수 있습니다. 다음은 MySQL 클라이언트에서 .sql 파일을 실행하는 방법입니다:

명령 프롬프트 (Windows) 또는 터미널 (Linux/Mac) 열기

MySQL 클라이언트 접속

MySQL 클라이언트에 접속하려면 다음 명령어를 입력합니다:

bash

코드 복사

mysql -u your\_username -p

여기서 your\_username은 MySQL의 사용자 이름입니다. 명령어를 입력한 후, 비밀번호를 입력하라는 메시지가 나타납니다.

데이터베이스 선택

SQL 파일의 쿼리가 적용될 데이터베이스를 선택합니다. 예를 들어, sample 데이터베이스를 선택하려면 다음 명령어를 입력합니다:

sql

코드 복사

USE sample;

SQL 파일 실행

다음 명령어를 입력하여 .sql 파일을 실행합니다:

sql

코드 복사

SOURCE /path/to/your\_file.sql;

여기서 /path/to/your\_file.sql은 SQL 파일의 경로입니다.

방법 2: MySQL Workbench에서 실행

MySQL Workbench와 같은 GUI 도구를 사용하는 경우, SQL 파일을 열고 실행할 수 있습니다:

MySQL Workbench 열기

SQL 파일 열기

MySQL Workbench에서 "File" 메뉴를 선택하고 "Open SQL Script"를 클릭하여 .sql 파일을 엽니다.

SQL 스크립트 실행

SQL 파일이 열린 후, 상단의 "Execute" 버튼을 클릭하여 스크립트를 실행합니다.

방법 3: 커맨드라인에서 직접 실행

커맨드라인에서 직접 SQL 파일을 실행할 수도 있습니다:

명령 프롬프트 (Windows) 또는 터미널 (Linux/Mac) 열기

다음 명령어를 입력하여 SQL 파일을 실행합니다:

bash

코드 복사

mysql -u your\_username -p your\_database < /path/to/your\_file.sql

여기서 your\_username은 MySQL의 사용자 이름, your\_database는 사용할 데이터베이스의 이름, /path/to/your\_file.sql은 SQL 파일의 경로입니다.

예시

SQL 파일을 C:\path\to\your\_file.sql에 저장했으며, MySQL 서버의 데이터베이스 sample에 적용하려면, 다음 명령어를 사용할 수 있습니다:

bash

코드 복사

mysql -u root -p sample < C:\path\to\your\_file.sql

이 명령어를 입력한 후 비밀번호를 입력하면 SQL 파일의 내용이 sample 데이터베이스에 적용됩니다.

요약

.sql 파일은 SQL 명령어를 포함하고 있으며, 데이터베이스에 적용하기 위해 실행해야 합니다.

MySQL 클라이언트, MySQL Workbench, 또는 커맨드라인에서 직접 실행할 수 있습니다.

데이터베이스에 적용하기 전에 SQL 파일의 내용을 확인하여 올바른 명령어가 포함되어 있는지 확인하는 것이 좋습니다.

V3

자료형 튜플로 변경

주요 수정 사항:

preprocess\_data 함수:

transactions를 튜플로 변환합니다. 리스트 대신 튜플을 사용하여 각 트랜잭션을 불변 데이터 구조로 만듭니다.

generate\_candidates 함수:

후보 생성 결과를 tuple로 반환합니다. set에서 후보를 생성하고 최종적으로 tuple로 변환하여 반환합니다.

apriori 함수:

itemset 및 frequent\_itemsets를 set으로 사용하여 중복을 자동으로 처리합니다. frequent\_itemsets를 업데이트한 후, tuple로 변환된 규칙을 반환합니다.

이렇게 수정된 코드는 튜플을 사용하여 데이터 구조의 불변성을 보장합니다.

v.5\_2 all

**변경사항 설명:**

1. **중간 그룹 추가**: current\_group이 업데이트될 때마다 그 상태를 related\_groups에 추가했습니다. 이로 인해 각 단계의 그룹을 모두 기록하게 됩니다.
2. **그룹 복사**: current\_group.copy()를 사용하여 현재 그룹의 상태를 복사하고 추가했습니다. 이를 통해 그룹의 상태가 계속 변경되더라도 이전 상태가 유지됩니다.

이제 코드가 중간 단계의 그룹도 포함하게 되어, 관련성 분석이 진행되는 모든 단계의 그룹을 추적할 수 있습니다.

Result:

[['action'], ['action', 'flow\_iat\_max'], ['action', 'flow\_iat\_mean', 'flow\_iat\_max'], ['action', 'flow\_iat\_mean', 'flow\_iat\_min', 'flow\_iat\_max'], ['action', 'flow\_iat\_max', 'flow\_iat\_min', 'flow\_iat\_mean', 'flow\_iat\_std'], ['flow\_iat\_total', 'action', 'flow\_iat\_max', 'flow\_iat\_min', 'flow\_iat\_mean', 'flow\_iat\_std'], ['flow\_iat\_total', 'action', 'flow\_iat\_max', 'flow\_iat\_min', 'flow\_iat\_mean', 'forward\_iat\_max', 'flow\_iat\_std'], ['flow\_iat\_max', 'flow\_iat\_mean', 'forward\_iat\_max', 'flow\_iat\_std', 'flow\_iat\_total', 'action', 'forward\_iat\_mean', 'flow\_iat\_min'], ['flow\_iat\_max', 'flow\_iat\_mean', 'forward\_iat\_max', 'forward\_iat\_min', 'flow\_iat\_std', 'flow\_iat\_total', 'action', 'forward\_iat\_mean', 'flow\_iat\_min'], ['flow\_iat\_max', 'forward\_iat\_std', 'flow\_iat\_mean', 'forward\_iat\_max', 'forward\_iat\_min', 'flow\_iat\_std', 'flow\_iat\_total', 'action', 'forward\_iat\_mean', 'flow\_iat\_min'], ['flow\_iat\_max', 'forward\_iat\_std', 'forward\_iat\_total', 'flow\_iat\_mean', 'forward\_iat\_max', 'forward\_iat\_min', 'flow\_iat\_std', 'flow\_iat\_total', 'action', 'forward\_iat\_mean', 'flow\_iat\_min'], ['backward\_iat\_max'], ['backward\_iat\_mean'], ['backward\_iat\_min'], ['backward\_iat\_std'], ['backward\_iat\_total'], ['backward\_packet\_length\_max'], ['backward\_packet\_length\_mean'], ['backward\_packet\_length\_min'], ['backward\_packet\_length\_std'], ['bpkts\_per\_second'], ['flow\_ack'], ['flow\_ack', 'infection'], ['flow\_cwr'], ['flow\_ece'], ['flow\_fin'], ['flow\_packets\_per\_second'], ['flow\_packets\_per\_second', 'flow\_iat\_max'], ['flow\_packets\_per\_second', 'flow\_iat\_min', 'flow\_iat\_max'], ['flow\_iat\_total', 'flow\_packets\_per\_second', 'flow\_iat\_min', 'flow\_iat\_max'], ['flow\_iat\_total', 'flow\_packets\_per\_second', 'flow\_iat\_max', 'flow\_iat\_min', 'flow\_protocol'], ['flow\_iat\_total', 'flow\_packets\_per\_second', 'flow\_iat\_max', 'flow\_iat\_min', 'flow\_protocol', 'forward\_iat\_max'], ['flow\_iat\_total', 'flow\_packets\_per\_second', 'flow\_iat\_max', 'flow\_iat\_min', 'flow\_protocol', 'forward\_iat\_max', 'forward\_iat\_min'], ['flow\_packets\_per\_second', 'flow\_iat\_max', 'forward\_iat\_total', 'forward\_iat\_max', 'forward\_iat\_min', 'flow\_iat\_total', 'flow\_iat\_min', 'flow\_protocol'], ['flow\_packets\_per\_second', 'flow\_iat\_max', 'forward\_iat\_total', 'forward\_iat\_max', 'forward\_iat\_min', 'flow\_iat\_total', 'flow\_iat\_min', 'flow\_protocol', 'forward\_packet\_length\_max'], ['flow\_packets\_per\_second', 'flow\_iat\_max', 'forward\_iat\_total', 'forward\_iat\_max', 'forward\_iat\_min', 'flow\_iat\_total', 'forward\_packet\_length\_mean', 'flow\_iat\_min', 'flow\_protocol', 'forward\_packet\_length\_max'], ['flow\_packets\_per\_second', 'flow\_iat\_max', 'forward\_iat\_total', 'forward\_iat\_max', 'forward\_iat\_min', 'flow\_iat\_total', 'forward\_packet\_length\_mean', 'forward\_packet\_length\_min', 'flow\_iat\_min', 'flow\_protocol', 'forward\_packet\_length\_max'], ['flow\_packets\_per\_second', 'flow\_iat\_max', 'fpkts\_per\_second', 'forward\_iat\_total', 'forward\_iat\_max', 'forward\_iat\_min', 'flow\_iat\_total', 'forward\_packet\_length\_mean', 'forward\_packet\_length\_min', 'flow\_iat\_min', 'flow\_protocol', 'forward\_packet\_length\_max'], ['flow\_packets\_per\_second', 'flow\_iat\_max', 'total\_bhlen', 'fpkts\_per\_second', 'forward\_iat\_total', 'forward\_iat\_max', 'forward\_iat\_min', 'flow\_iat\_total', 'forward\_packet\_length\_mean', 'forward\_packet\_length\_min', 'flow\_iat\_min', 'flow\_protocol', 'forward\_packet\_length\_max'], ['flow\_packets\_per\_second', 'flow\_iat\_max', 'total\_bhlen', 'fpkts\_per\_second', 'forward\_iat\_total', 'forward\_iat\_max', 'forward\_iat\_min', 'total\_forward\_packets', 'flow\_iat\_total', 'forward\_packet\_length\_mean', 'forward\_packet\_length\_min', 'flow\_iat\_min', 'flow\_protocol', 'forward\_packet\_length\_max'], ['total\_length\_of\_forward\_packets', 'flow\_packets\_per\_second', 'flow\_iat\_max', 'total\_bhlen', 'fpkts\_per\_second', 'forward\_iat\_total', 'forward\_iat\_max', 'forward\_iat\_min', 'total\_forward\_packets', 'flow\_iat\_total', 'forward\_packet\_length\_mean', 'forward\_packet\_length\_min', 'flow\_iat\_min', 'flow\_protocol', 'forward\_packet\_length\_max'], ['flow\_psh'], ['forward\_packet\_length\_std', 'flow\_psh'], ['flow\_rst'], ['flow\_syn'], ['reconnaissance', 'flow\_syn'], ['flow\_urg'], ['total\_backward\_packets'], ['total\_fhlen'], ['total\_length\_of\_backward\_packets']]

Element count 1: 23 groups

Element count 2: 5 groups

Element count 3: 2 groups

Element count 4: 2 groups

Element count 5: 2 groups

Element count 6: 2 groups

Element count 7: 2 groups

Element count 8: 2 groups

Element count 9: 2 groups

Element count 10: 2 groups

Element count 11: 2 groups

Element count 12: 1 groups

Element count 13: 1 groups

Element count 14: 1 groups

Element count 15: 1 groups