|  |
| --- |
| 1102040007 오혁진  1102340003 유자연  1102378004 서영재  1102378010 강민정 |

|  |
| --- |
| 숭실대학교 |
| **인공지능특론1 보고서** |
| **Assignment #1 – Anonymization** |

목차

[1 서론 - 1 -](#_Toc131542028)

[2 익명화 - 3 -](#_Toc131542029)

[2.1 데이터 선정 - 3 -](#_Toc131542030)

[2.2 익명화 방법 - 4 -](#_Toc131542031)

[3 익명화 결과 - 8 -](#_Toc131542032)

[3.1 익명화 데이터 - 8 -](#_Toc131542033)

[3.2 익명화 검증 - 11 -](#_Toc131542034)

[4 결론 - 12 -](#_Toc131542035)

1. 서론

* 개념

K-anonymity : 데이터 세트의 각 그룹(또는 카테고리)이 동일한 속성 세트를 가진 k 개   
이상의 개인을 갖도록 요구하는 프라이버시 개념

L-diversity : 데이터 세트의 각 그룹(또는 카테고리)이 민감한 속성에 대해 최소한 *l* 개의 고유한 값을 가져야 하는 개인 정보 보호

T-closeness : 그룹(또는 카테고리)의 민감한 속성 분포와 데이터 세트의 민감한 속성의   
전체 분포 사이의 거리를 측정하는 프라이버시 개념

예시)

K-anonymity :

|  |  |  |
| --- | --- | --- |
| 나이 | 성별 | 건강 상태 |
| 32 | 남 | 당뇨 |
| 32 | 남 | 암 |
| 24 | 여 | 암 |
| 24 | 여 | 당뇨 |

* 남성 환자 모두와 여성 환자 모두는 암과 당뇨를 앓고 있음
* 이 데이터 세트는 연령, 성별 및 질병 속성과 관련하여 k = 2 를 충족

L-diversity :

|  |  |  |
| --- | --- | --- |
| 거래 금액(만 원) | 위치 | 카드 소유자 |
| 50 | 서울 | 김철수 |
| 50 | 부산 | 이영희 |
| 75 | 대전 | 김철수 |
| 75 | 대구 | 이영희 |

* 50만 원, 75만 원 거래는 각 서로 다른 개인이 거래함
* 이 데이터 세트는 거래 금액 속성에 대해 l = 2 를 만족

T-closeness :

|  |  |  |
| --- | --- | --- |
| 나이 | 성별 | 등급 |
| 22 | 남 | 4 |
| 24 | 여 | 2 |
| 38 | 남 | 5 |
| 40 | 여 | 3 |

|  |  |  |
| --- | --- | --- |
| 나이 | 성별 | 등급 |
| 20-30 | 남 | 4 |
| 20-30 | 여 | 2 |
| 30-40 | 남 | 5 |
| 30-40 | 여 | 3 |

* 각 그룹의 등급 데이터 세트의 전체 등급 분포와 크게 다르지 않도록 등급을 그룹화해야함
* 위 예에서 각 그룹의 등급 분포는 크게 다르지 않음

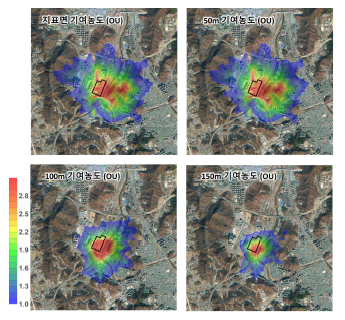
1. 익명화
   1. 데이터 선정

데이터 : 종관기상관측(23년 2월 1일 24시간 분당 데이터) 중 7 개 지역   
임의 선택

내용 : 측정 지점, 일시, 기온(°C), 풍향(deg), 풍속(m/s), 현지기압(hPa), 습도(%) 등의 데이터

데이터 수 : 약 1 만 개

데이터 사용 예시 : 시뮬레이션 프로그램을 이용한 특정 지역의 시간에 따른   
대기오염물질 확산 추이 예상



[출처] 2023 악취 및 대기확산 모델링(CALPUFF) 교육, 안양대학교 기후**·**에너지**·**환경융합연구소, 한국냄새환경학회

커스터마이징 : 해당 데이터에 특정 민감 정보가 존재하지 않아 임의로 설정  
(QI : “지점”, “일시”, “기온(°C)” / SA : “풍향(deg)”)

**※ k, l, t 값에 임의의 값을 넣기 위해 칼럼별로 일 또는 십의 자리에서 반올림하여 데이터 수정**

* 1. 익명화 방법

익명화 값 :

**k-anonymity 🡺 k = 5**

**l-diversity 🡺 l = 3**

**t-closeness 🡺 t = 0.2**

익명화 파이썬 코드(data 저장 코드는 보고서에서는 생략)

|  |
| --- |
| # pip install openpyxl  # pip install faker  # pip install pycanon  # pip install matplotlib  #!pip install pycanon  import pandas as pd  from pycanon import anonymity, report  df = pd.read\_csv( "./original\_data(2).csv")  df.head() |

* **사전처리**
* **k-Anonymity**

|  |
| --- |
| import numpy as np  def generalize(data, columns, generalization\_levels):      generalized\_data = data.copy()      for col, level in zip(columns, generalization\_levels):          if level > 0 and np.issubdtype(data[col].dtype, np.number):              col\_min = data[col].min()              col\_max = data[col].max()              step = (col\_max - col\_min) // level              generalized\_data[col] = (data[col] // step) \* step      return generalized\_data  def check\_k\_anonymity(df, columns, k):      grouped\_data = df.groupby(columns).size().reset\_index(name='count')      return all(grouped\_data['count'] >= k)  def k\_anonymize(data, columns, k, generalization\_levels):      generalized\_data = generalize(data, columns, generalization\_levels)      if check\_k\_anonymity(generalized\_data, columns, k):          return generalized\_data      else:          print("The dataset doesn't satisfy k-anonymity.")          return None  # Load the CSV file  file\_path = "./original\_data(2).csv"  data = pd.read\_csv(file\_path)  # Define the columns to anonymize and their generalization levels  # Here, we assume that you have 3 columns to anonymize, and the generalization levels are [2, 2, 2]  columns\_to\_anonymize = ["현지기압(hPa)"  ,"해면기압(hPa)"]  generalization\_levels = [5]  # Apply k-anonymity  k = 5  anonymized\_data = k\_anonymize(data, columns\_to\_anonymize, k, generalization\_levels)  if anonymized\_data is not None:      print("Anonymized data:")      print(anonymized\_data) |

* **l-diversity**

|  |
| --- |
| def check\_l\_diversity(df, sensitive\_column, columns, l):      grouped\_data = df.groupby(columns)[sensitive\_column].nunique().reset\_index(name='unique\_values')      return all(grouped\_data['unique\_values'] >= l)  def l\_diversify(data, sensitive\_column, columns, l):      if check\_l\_diversity(data, sensitive\_column, columns, l):          return data      else:          print("The dataset doesn't satisfy l-diversity.")          return None  # Load the CSV file  file\_path\_2="./Anonymized\_data\_output1.csv"  data\_2 = pd.read\_csv(file\_path\_2)  # Define the sensitive column and the columns to group by (e.g., anonymized columns)  sensitive\_column = '지점'  columns\_to\_group\_by = ['풍향(deg)']  # Apply l-diversity  l = 3  diversified\_data = l\_diversify(data\_2, sensitive\_column, columns\_to\_group\_by, l)  if diversified\_data is not None:      print("Diversified data:")      print(diversified\_data) |

* **t-closeness**

|  |
| --- |
| from scipy.stats import wasserstein\_distance  def get\_distribution(df, sensitive\_column):      return df[sensitive\_column].value\_counts(normalize=True)  def apply\_t\_closeness(df, sensitive\_column, columns, t):      overall\_dist = get\_distribution(df, sensitive\_column)      grouped\_data = df.groupby(columns)      result\_df = pd.DataFrame()      for \_, group in grouped\_data:          group\_dist = get\_distribution(group, sensitive\_column)          emd = wasserstein\_distance(overall\_dist, group\_dist)          if emd <= t:              result\_df = result\_df.append(group)      if not result\_df.empty:          return result\_df.reset\_index(drop=True)      else:          print("No groups satisfy t-closeness.")          return None  # Load the CSV file  file\_path = "./original\_data(2).csv"  data = pd.read\_csv(file\_path)  # Define the sensitive column and the columns to group by (e.g., anonymized columns)  sensitive\_column = '지점'  columns\_to\_group\_by = ['풍향(deg)','풍속(m/s)']  # Apply k-anonymity and l-diversity (or any other anonymization techniques)  # ...  # Apply t-closeness  t = 0.2  tclosed\_data = apply\_t\_closeness(data, sensitive\_column, columns\_to\_group\_by, t)  if tclosed\_data is not None:      # Print the DataFrame      print("Data after applying t-closeness:")      print(tclosed\_data) |

1. 익명화 결과
   1. 익명화 데이터

* K-anonymity 데이터 비교



* K-anonymity + L-diversity





* T-closeness





* 1. 익명화 검증
* K-anonymity + L-diversity

텍스트, 편지이(가) 표시된 사진

자동 생성된 설명

* T-closeness

텍스트, 편지이(가) 표시된 사진

자동 생성된 설명

1. 결론

k-anonymity, l-diversity, t-closeness 의 서로 다른 k, l , t 값을 이용하여 익명화된 데이터 생성

* k, t 에 대한 값들은 정상적으로 적용하여 진행
* 처음에는 l 값은 임의로 다른 값을 넣는 것에 어려움이 있었음
* 기존의 데이터를 그대로 쓰는 것이 아닌 불필요한 칼럼 제거 및 데이터 반올림(일의 자리 또는 십의 자리에서)하여 데이터 커스터마이징 후 적용

pyCanon을 사용하여 익명성 검증

* pyCanon을 사용하여 진행
* k, t, l에 대한 값들은 정상적으로 report가 생성됨을 확인
* 모든 값들이 각 값들의 정확한 값이 report 된 것은 아님
* 정확한 값 또는 근사치의 값이 report 되는 것을 확인