SDVmetrics를 이용한 재현자료 평가방법 소개

김정훈

2023-07-25

목차

1. 재현자료란 무엇인가?

1. 재현자료란 무엇인가?

- 재현자료란 실제로 측정하지 않은 임의의 데이터로 넓게 정의하기도 하지만, 통상적으로 추정된 모형에서 생성된 가상의 데이터를 의미한다.
- 실제로 공공기관에서 제공하는 데이터들은 재현자료일 경우가 많다. 보안성의 이유 때문에 재현자료가 필수적임
- ① **완전 재현 자료**는 실제데이터가 하나도 없이 가상으로만 이루어진 데이터를 의미하며, 정보보호 측면에서 가장 강력한 보안성을 가진다.
- ② **부분 재현 자료**는 공개하려는 변수들 중 일부만 선택하여 재현자료로 대체한 데이터를 의미하며, 보통 재현자료로 대치되는 변수들은 민감한 정보에 관현 변수들이다.
- ③ **복합 재현자료**는 일부 변수들의 값을 재현자료로 생성하고, 생성된 재현자료와 실제데이터를 모두 이용하여 일부 변수 값들을 다시 도출하는 방법으로 생성한다.

2. CTGAN의 간단한 소개

2. CTGAN의 간단한 소개

1. 모드별 정규화 (Mode-Specific Normalization)
2. 조건부 적대적 생성망(Conditional GAN)
    1) 조건 벡터(Conditional Vector)
    2) 생성자 손실 함수(Generator Loss Function)
    3) 샘플링 훈련 전략(Training-by-sampling)
    4) 신경망 구조(Neural Network Structure)

2. CTGAN의 간단한 소개

1. 모드별 정규화 (Mode-Specific Normalization)
   - 원래는 (최소 - 최대 정규화 : 최대값 1 , 최소값 -1로 설정) 하는 방법을 사용했지만, 수치형 데이터가 가우시안 분포가 아닌 다른 분포의 형태를 가질 때, 역전파 과정에서 Gradient Vanishing이 발생할 수 있다.
   - GAN 모델이 다중 모드를 잘 발견하지 못하는 문제 발생

위와 같은 문제를 해결하기 위해 일반적인 형태의 분포를 혼합 가우시안 형태로 나타내고 특정 레코드의 모드를 명시적으로 나타내도록 전처리 이는 곧, **변분 가우시안 혼합 모델**이라고 말하며 이를 통해 정규화를 진행

2. CTGAN의 간단한 소개

변분 가우시안 혼합 모델

①. k개의 혼합분포를 찾은 다음에 가중치(μ_k)가 특정 임계치 ϵ 이상인 경우의 분포를 남긴다.
$\Rightarrow \mathbb{P}_{C_{i,j}} = \Sigma_{k=1}^{3} \mu_k N(c_{i,j}; \eta_k, \phi_k)$
여기서 μ_k,η_k, ϕ_k는 모드의 평균, 가중치 및 표준편차이다.

②. 특정값 $c_{i,j}$에 대해 각 가우시안 분포에서의 확률 $\mu_k N(c_{i,j}; \eta_k, \phi_k)$
$p(k) = \frac{\mu_k N(c_{i,j};\eta_k,\phi_k)}{\Sigma_p \mu_k N(c_{i,j};\eta_p,\phi_p)}$확률을 이용하여서 $C_{i,j}$가 어떤 가우시안 분포에서 나왔는지 선택

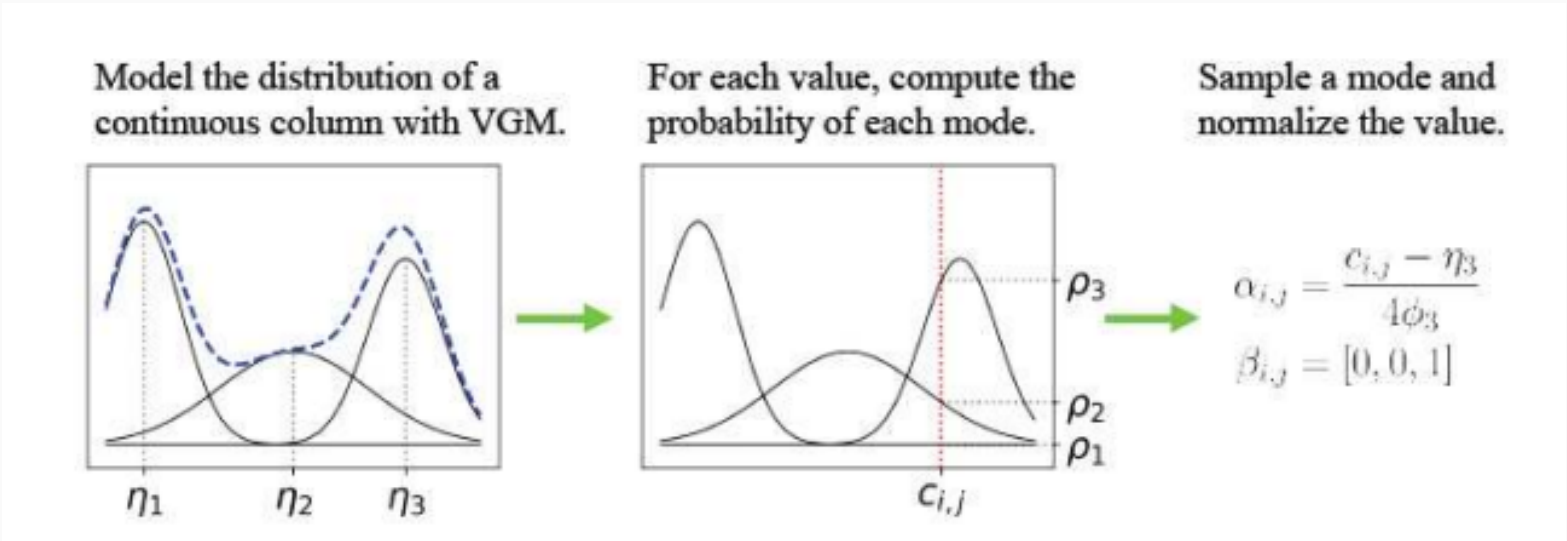
ex) $p_1, p_2, p_3 = [0.1 , 0.3 , 0.1]$ 이라고 했을때 1번이 20% , 2번이 60% , 3번이 20%로 선택될 확률

2. CTGAN의 간단한 소개

③. 위 단계에서 선택된 가우시안 분포의 평균과 표준편차를 이용, 값 $c_{i,j}$를 $a_{i,j}$로 정규화, 어떤 정규분포가 선택되었는지 원핫벡터 $\beta_{i,j}$로 표시

④. j번째 레코드는 수치형 값 $c_{i,j}$와 범주형 값 원핫벡터 ($d_{i,j}$)로 구성된 형태에서 모드별 정규화를 이용. 수치형 데이터 ($\alpha_{i,j}, \beta_{i,j}$)와 범주형 데이터 원핫 벡터 ($d_{i,j}$)로 구성된 형태.

$\Rightarrow r_j = \alpha_{i,j} \oplus \beta_{i,j} \oplus ... \oplus d_{i,j} \oplus ...d_{N_d,j}$



2. CTGAN의 간단한 소개

2. 조건부 적대적 생성망 (Conditional GAN)

적대적 생성망에서의 범주형 자료에서 특정 클래스 희소성을 해결하기 위해 사용하며 3가지 조건을 만족해야한다

1) 조건벡터
2) 생성자 손실 함수(Generator Loss Function)
3) 샘플링 훈련 전략(Training-by-sampling)

위 문제들을 해결하면 생성한 재현자료의 조건부 분포($P_g(row|D_i* = k)$는 실제 데이터 분포의 조건부 분포($P(row|D_i* = k)$)와 같아지고 다음 식을 통해 실제 데이터 분포를 얻을 수 있게 된다.

$$\mathbb{P}(row) = \sum_{k \in D_i*} \mathbb{P}_g(row|D_{i*} = k*)\mathbb{P}(D_{i*} = k*)$$

2. CTGAN의 간단한 소개

① 조건 벡터

- 범주형 데이터 열에 대한 조건을 만들기 위해 원핫벡터 형태의 마스킹 벡터를 이용
- 범주형 열 중 하나의 열만 선택하고 그 열의 여러 클래스 중 하나의 클래스만 선택해서 조건으로 나타냄.
- 선택된 클래스 일 때 1로 , 나머지는 0으로 표현
- 조건벡터는 마스킹벡터를 모두 결합한 형태인 $m_1 \oplus ... \oplus m_{N_d}$ (여기서 N_d는 범주형 데이터 열의 갯수로 표현)

ex) 2번째 열에서 1이 선택되었다면, $m_1 = [0, 0, 0], m_2 = [1, 0]$ 이면 조건벡터는 [0,0,0,1,0] 이 선택됨

2. CTGAN의 간단한 소개

② 생성자 손실 함수

- 위의 예시처럼 [0,0,0,1,0] 조건이 들어왔다면, 첫번째 범주형 열은 아무 클래스나 들어가 생성해도 되지만 두번째 범주형 열은 1이라는 값이 생성되어야 한다.
- 즉, 생성된 두번째 범주형 열 데이터($\hat{d}_2$)가 조건(m_2)와 같아야한다.
- 생성자가 조건과 같은 클래스를 만들도록 학습시키기 위해 생성자 손실 함수에 $\hat{d}_2$ 와 m_2의 크로스 엔트로피 손실을 추가한다.
- $CrossEntropy = \sum_{k=1}^{|D_i|} m_i^{(k)} log \hat{d}_i^{(k)}$ 여기서 ($|D_i|$는 i번째 범주형 열의 클래스 갯수)

2. CTGAN의 간단한 소개

③ 샘플링 훈련 전략

- 생성자가 만든 재현자료의 조건부 분포 $=$ 실제 데이터의 조건부 분포 $\rightarrow$ 식별자가 두 분포간의 거리를 정확하게 추정
- 모든 사용 가능한 조건벡터와 훈련 데이터를 사용해야만 제대로 된 학습이 가능하다. $\rightarrow$ 균일하게 조건 벡터와 훈련 데이터를 표본을 뽑는 전략을 통해 위 문제를 해결 가능.
- 구체적인 방법은 6단계로 나누어서 설명.

2. CTGAN의 간단한 소개

① 범주형 열 중 하나를 균등한 확률로 선택

② 선택된 범주형 열에서 각 값의 발생빈도를 통해 확률 분포를 만든다
이 때 각 빈도에는 로그함수를 취해준다.

ex) D_2에서 1번째 클래스가 100번 발생하고, 두번째 클래스가 50번 발생했다면, 확률분포는

$(\frac{log100}{log100+log50}, \frac{log50}{log100+log50}) = (0.54\ ,\ 0.46)$

③ 2번에서 구한 확률분포에 따라 1개의 값을 선택, 위의 그림을 봤을 때는 첫번째 클래스가 선택되었다.

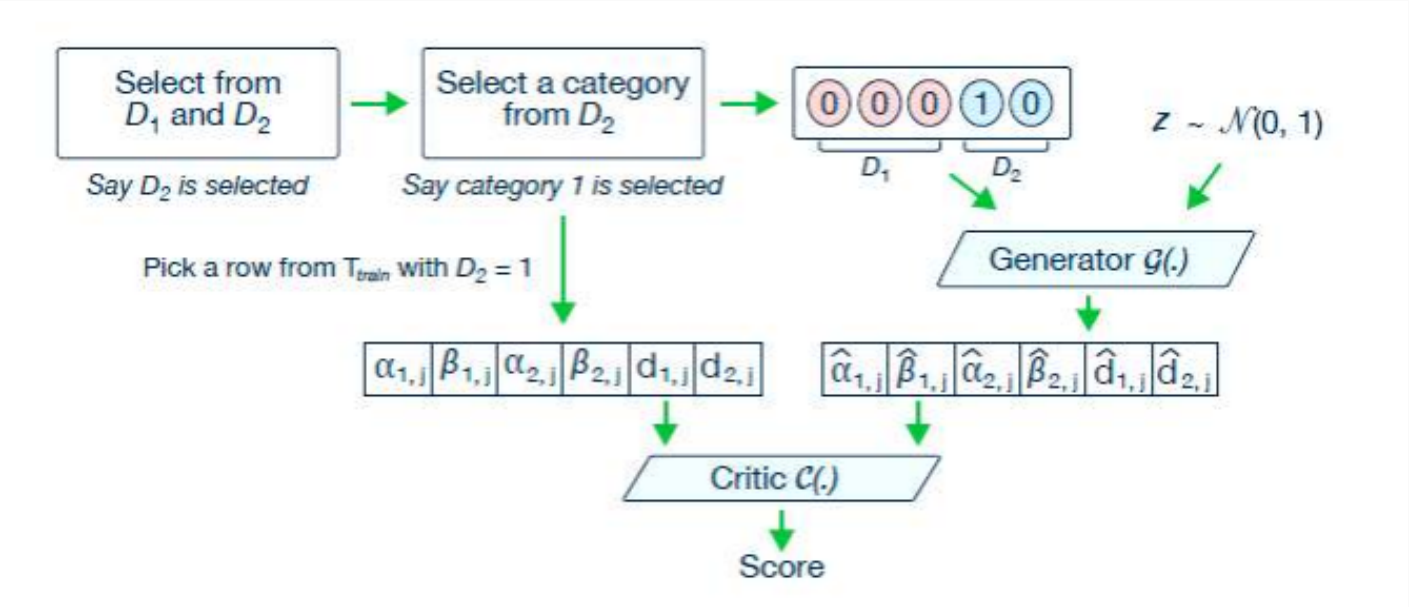
④ 선택된 열과 클래스에 따라 조건 벡터를 만들고 훈련데이터를 랜덤추출한다.

ex) 위의 그림을 참조 했을 때 범주형 열 D_1(클래스 3개) , D_2(클래스 2개) 중에서 D_2의 첫번째 클래스가 선택되었으므로 $m_1 = [0,0,0]$, $m_2 = [1,0]$, 조건벡터는 [0,0,0,1,0]이다.

2. CTGAN의 간단한 소개

⑤ 생성자에 조건벡터와 다변수 가우시안 분포에서 추출한 잠재변수를 입력변수로 넣어서 재현자료를 생성한다.

⑥ 식별자에 실제 데이터와 조건벡터, 재현자료와 조건벡터를 각각 넣어서 나온 결과로 두 분포간의 거리(Score)를 계산해 식별자와 생성자를 업데이트한다. 식별자는 거리가 멀어지게, 생성자는 거리가 가까워지게 하는 방향으로 학습



2. CTGAN의 간단한 소개

신경망 구조

① 생성자 신경망 구성

$$
\begin{cases}
h_0 = z \oplus cond \\
h_1 = h_0 \oplus ReLU\left(BN\left(FC_{|cond|+|z| \to 256}\left(h_0\right)\right)\right) \\
h_2 = h_1 \oplus ReLU\left(BN\left(FC_{|cond|+|z|+256 \to 256}\left(h_1\right)\right)\right) \\
\widehat{\alpha}_i = tanh\ \left(FC_{|cond|+|z|+512 \to 1}\left(h_2\right)\right) & 1 \le i \le N_c \\
\widehat{\beta}_i = gumbel_{0.2}\left(FC_{|cond|+|z|+512 \to m_i}\left(h_2\right)\right) & 1 \le i \le N_c \\
\widehat{d}_i = gumbel_{0.2}\left(FC_{|cond|+|z|+512 \to |D_i|}\right)
\end{cases}
$$

2. CTGAN의 간단한 소개

신경망 구조

② 식별자 신경망 구성

$$
\begin{cases}
h_0 = r_1 \oplus \ldots \oplus r_{10} \oplus cond_1 \oplus \ldots \oplus cond_{10} \\
h_1 = drop\left(leaky_{\ 0.2}\left(FC_{10|r|+10|cond| \to 256}\left(h_0\right)\right)\right) \\
h_2 = drop\left(leaky_{\ 0.2}\left(FC_{256 \to 256}\left(h_1\right)\right)\right) \\
C(\bullet) = FC_{256 \to 1}\left(h_2\right)
\end{cases}
$$

3. 데이터 소개

3. 데이터 소개

본 데이터는 2016 가계 금융 복지 조사의 가구마스터 자료이며, 총 레코드의 숫자는 18,273개의 레코드이며, 편의성 가중치는 고려하지 않았다.

- 범주형 변수 : 수도권 여부 , 가구주 성별 , 학교(학력코드) , 연령
  - 수도권 여부(urban) : G1 (수도권) , G2 (비수도권)
  - 가구주 성별(sex) : 1 (남자) , 2 (여자)
  - 학력코드(edu) : 1 (안받음) , 2 (초등학교) , 3 (중학교) , 4 (고등학교) , 5 (대학 (3년제 이하)) , 6 (대학교 (4년제이상)) , 7 (대학원 이상)
  - 가구주 나이(age) : G1 (30세 미만) , G2 (30 40세 미만) , G3 (40 50세 미만) , G4 (50 60세 미만) , G5 (60세이상)

3. 데이터 소개

- 연속형 변수 : 경상소득 , 근로소득 , 사업소득 , 재산소득

  - 경상소득 : 비교적 오랫동안 정기적으로 얻는 소득
    (경상소득 = 근로소득 + 사업소득 + 재산소득 + 공적이전소득 + 사적이전소득)

  - 근로소득(labor) : 사업체에 고용되어 근로를 제공한 대가로 받은 모든 현금과 현물을 의미한다. 여기에서 현물이란가구소득 정의에서와 마찬가지로 재화와 서비스를 포함하는 개념.

  - 사업소득(business) : 비법인기업의 주인이 해당 사업체를 운영하여 얻은 순수입이다. 여기에서 순수입이란 총수입액 또는 총매출액에서 영업비용 등 생산에 사용한 생산비용을 제외한 금액을 의미한다.

  - 재산소득(property) :소유한 재산을 타인이 사용한 대가로 받은 순수입이다. 여기에는 임대소득, 이자소득, 배당소득, 연금소득, 상표 사용료/인세/사용료 등이 포함된다.

4. 재현자료 생성과정

4. 재현자료 생성과정

재현자료는 CTGAN, CTGANSynthesizer를 통해 생성할 수 있으며,
CTGAN의 경우 SDV의 구버전으로, CTGANSynthesizer는 SDV의
신버전으로 생성 할 수 있으며,
지금까지 해본 결과 구버전이 신버전보다 속도가 더 빠르고,
정확도면에서도 신버전이 더 느리다고 구버전보다 유의미한 퍼포먼스는
보여주지 않았다.

① SDV version = 0.14 (구버전)

② SDV version = 1.0.0 (신버전)

4. 재현자료 생성과정

① SDV version = 0.14

```
## 패키지 불러오기
import sdv
from sdv.tabular import CTGAN

## 실행 코드
model = CTGAN(primary_key='ID' , epochs = 30 , batch_size = 400 , verbose = True)
model.fit(data)


## 모델 저장 및 불러오기
model.save('Path/file_name.pkl')
model.load('Path/file_name.pkl')

## 재현 데이터 생성
model.sample(len(data))
```

4. 재현자료 생성과정

② SDV version = 1.0.0

```
## 패키지 불러오기
import sdv
from sdv.metadata import SingleTableMetadata
from sdv.single_table import CTGANSynthesizer

## 실행 코드
metadata = SingleTableMetadata()
metadata.detect_from_dataframe(data = data)
synthesizer = CTGANSynthesizer(
    metadata, # required
    enforce_rounding=False,
    epochs=30,
    batch_size = 400,
    verbose=True)
synthesizer.fit(data)

## 재현 데이터 생성
synthetic_data = synthesizer.sample(num_rows = len(data) , batch_size = 400)
```

5. 재현자료의 평가방법

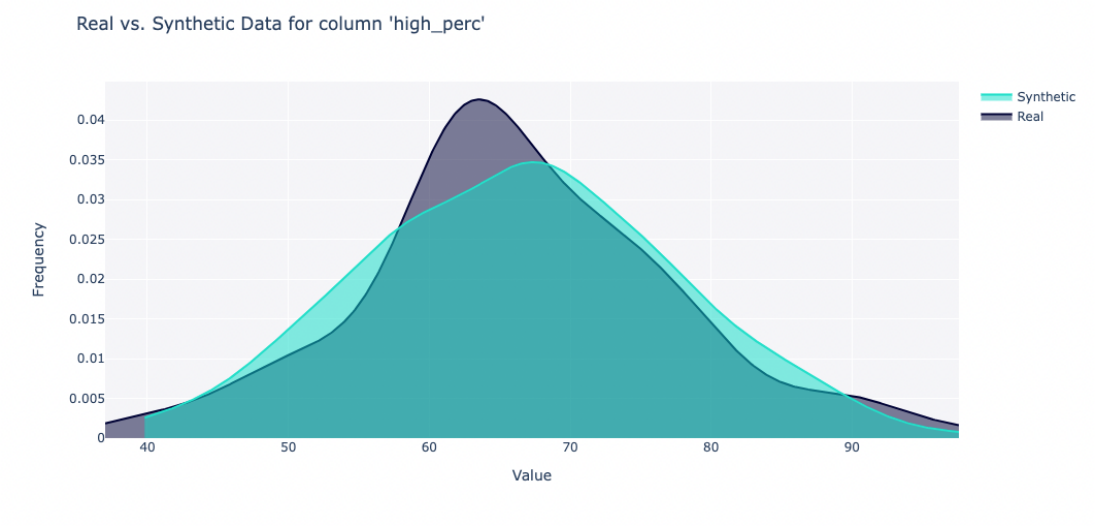
5. 재현자료의 평가방법

- CTGAN을 통해 재현자료를 생성하게되면 재현자료가 얼마나 원자료와 유사하게 나왔는지를 알아보고싶을 것이다.
- SDMetrics에서 SingleTable의 경우와 MultiTable의 경우를 나눠서 평가방법들을 소개해 놓았다.
- 그래프나 단순척도로 표현할 수 있어 매우 좋은 패키지라 생각되어서 소개하는 글을 쓰게 되었다.
- 싱글데이터의 범위 내에서 평가 방법을 정리한다.

5. 재현자료의 평가방법

1. Quality Report

열 모양 , 열 쌍 추세 및 테이블 관계를 보여주고 전역 품질 지표를 보여준다.



- Numerical, Datetime → KSComplement
- Boolean, Categorical → TVComplement

5. 재현자료의 평가방법

① KSComplement

측도 KSComplement는 열 모양 (열의 한계 분포 또는 1D 히스토그램) 측면에서 실제자료의 열과 재현자료 열의 유사성을 계산한다.

1. Numerical : 이 메트릭은 연속적인 수치 데이터를 의미한다.
2. Datetime : 이 메트릭은 날짜 시간의 값을 숫자 값으로 변환한다.

※ 결측값은 무시함.

점수는 0.0 ~ 1.0의 형태로 보여주고 1.0으로 갈수록 실제자료와 재현자료의 유사성이 높은것으로 판단하면 된다.

5. 재현자료의 평가방법

실행 코드

```
from sdmetrics.single_column import KSComplement

KSComplement.compute(
    real_data=real_table['column_name'],
    synthetic_data=synthetic_table['column_name']
)
```

- 파라미터
  - real_data : 실제자료에서의 한개의 연속형 Column
  - synthetic_data : 재현자료에서의 한개의 연속형 Column

5. 재현자료의 평가방법

② TVComplement

측도 TVComplement는 열 모양 (열의 한계 분포 또는 1D 히스토그램) 측면에서 실제자료의 Column 대 재현자료의 Column의 유사성을 계산한다.

1. Categorical : 범주형 자료 형태의 Column
2. Boolean : True , False 형태의 Column

※ 결측값은 무시함.

점수는 0.0 ∼ 1.0의 형태로 보여주고 1.0으로 갈수록 실제자료와 재현자료의 유사성이 높은것으로 판단하면 된다.

5. 재현자료의 평가방법

실행 코드

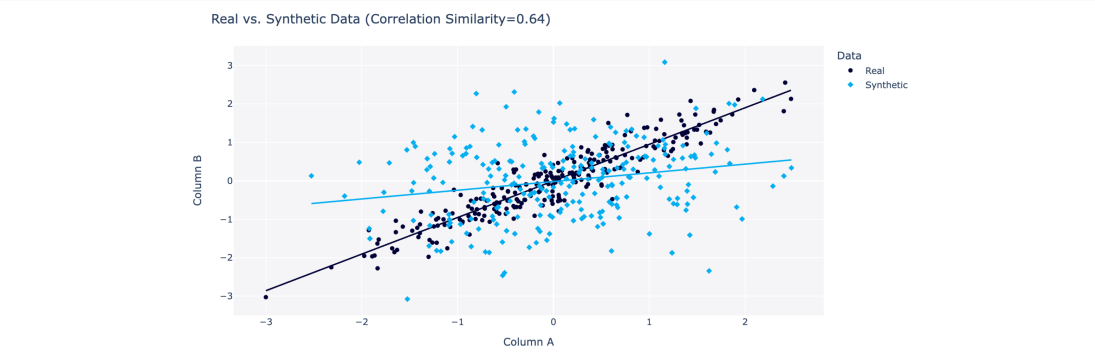
```
from sdmetrics.single_column import TVComplement

TVComplement.compute(
    real_data=real_table['column_name'],
    synthetic_data=synthetic_table['column_name']
)
```

- 파라미터
  - real_data : 실제자료에서의 한개의 범주형 Column
  - synthetic_data : 재현자료에서의 한개의 범주형 Column

5. 재현자료의 평가방법

2. Column Pair Trends

재현자료와 실제자료의 추세를 확인해보고 싶을때 상관관계와 같은 척도를 이용하여 비교한다.



- Numerical with another Numerical (including datetime) → CorrelationSimilarity
- Categorical with another Categorical → ContingencySimilarity
- Numerical with a Categorical → 수치화 열을 빈도수로 이산화 한 후 ContingencySimilarity 적용

5. 재현자료의 평가방법

① CorrelationSimilarity

측도 CorrelationSimilarity는 한 쌍의 숫자 열 사이의 상관 관계를 측정하고 실제자료와 합성 데이터 간의 유사성을 계산한다.
즉, 2D 분포의 추세를 비교한다. 이 메트릭은 상관 관계를 측정하기 위해 Pearson 및 Spearman의 상관계수를 이용 할 수 있다.

1. Numerical : 이 메트릭은 연속적인 수치 데이터를 의미한다.
2. Datetime : 이 메트릭은 날짜 시간의 값을 숫자 값으로 변환한다.

점수는 0.0 $\sim$ 1.0의 형태로 보여주고 1.0으로 갈수록 실제자료와 재현자료의 유사성이 높은것으로 판단하면 된다.

5. 재현자료의 평가방법

실행 코드

```
from sdmetrics.column_pairs import CorrelationSimilarity

CorrelationSimilarity.compute(
    real_data=real_table[['column_1', 'column_2']],
    synthetic_data=synthetic_table[['column_1', 'column_2']],
    coefficient='Pearson'
)
```

- 파라미터
  - real_data : 실제자료에서의 한개의 연속형 Column
  - synthetic_data : 재현자료에서의 한개의 연속형 Column
  - coefficient : 피어슨 상관계수를 이용할지, 스피어만 상관계수를 이용할지

5. 재현자료의 평가방법

② ContingencySimilarity

측도 ContingencySimilarity는 실제자료의 세트와 재현자료 세트 사이의 범주형 열 쌍의 유사성을 계산한다. 즉, 2D 분포를 비교한다.
즉, 2D 분포의 추세를 비교한다. 이 메트릭은 상관 관계를 측정하기 위해 Pearson 및 Spearman의 상관계수를 이용 할 수 있다.

1. Categorical : 범주형 자료 형태의 열
2. Boolean : True , False 형태의 열

※ 이 메트릭을 사용하려면 두 열이 모두 호환되어야 한다. 열에 결측값이 있으면 메트릭은 이 값을 추가 단일 범주로 처리한다.

점수는 0.0 ∼ 1.0의 형태로 보여주고 1.0으로 갈수록 실제자료와 재현자료의 유사성이 높은것으로

판단하면 된다.

5. 재현자료의 평가방법

실행 코드

```python
from sdmetrics.column_pairs import ContingencySimilarity

ContingencySimilarity.compute(
    real_data=real_table[['column_1', 'column_2']],
    synthetic_data=synthetic_table[['column_1', 'column_2']]
)
```

- 파라미터
  - real_data : 실제자료에서의 두개의 범주형 Column
  - synthetic_data : 재현자료에서의 두개의 범주형 Column

5. 재현자료의 평가방법

③ Quality Report의 세부 function들

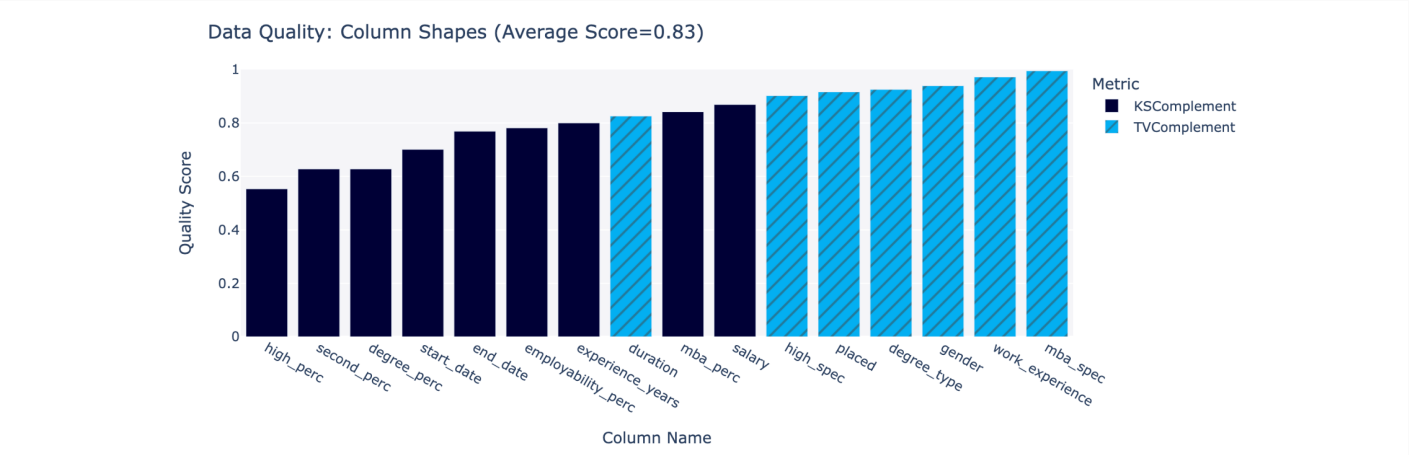
싱글 테이블의 경우이므로 싱글 테이블에서 선언해준다.

1. generate(real_data, synthetic_data, metadata , verbose)
2. get_score() : 전체적인 재현자료와 실제자료간의 유사도 점수를 보고자 할 때 (0 ~ 1 사이 값)
3. get_properties() : 이 메트릭이 측정한 각각의 특성을 보고자 할 때
4. get_details() : 어떤 특정한 열에 대해서 특성을 알고 싶을 때

```
from sdmetrics.reports.single_table import QualityReport

report = QualityReport()

report.generate(real_data, synthetic_data, metadata)
report.get_score()
report.get_properties()
report.get_details(property_name='Column Shapes')
fig = report.get_visualization(property_name='Column Shapes')
fig.show()
```

5. 재현자료의 평가방법

- property_name : 'Column Shapes' or 'Column Pair Trends'



5. 재현자료의 평가방법

3. Diagnostic Report

싱글 테이블의 경우이므로 싱글 테이블에서 선언해준다.

1. generate(real_data, synthetic_data, metadata , verbose)
2. get_results()
3. get_properties()
4. get_details(property_name)

```
from sdmetrics.reports.single_table import DiagnosticReport

report = DiagnosticReport()

report.generate(real_data, synthetic_data, metadata)
report.get_results()
report.get_properties()
report.get_details(property_name='Coverage')
fig = report.get_visualization(property_name='Coverage')
fig.show()
```

5. 재현자료의 평가방법

4. Visualization Utilities

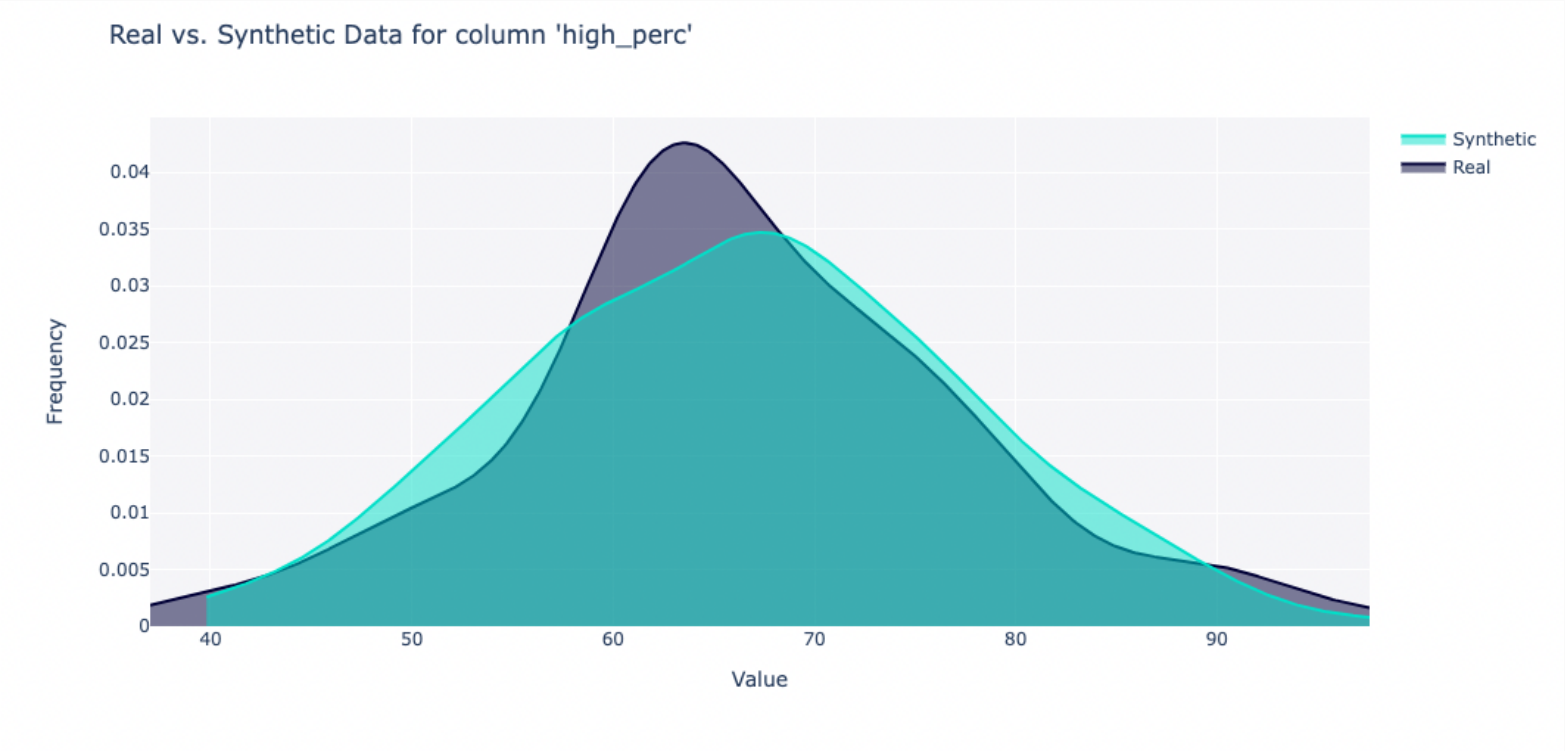
1. 1D형태로 실제자료와 재현자료의 분포 표현

```
from sdmetrics.reports import utils

fig = utils.get_column_plot(
    real_data=real_table,
    synthetic_data=synthetic_table,
    column_name='high_perc',
    metadata=my_table_metadata_dict
)

fig.show()
```

- 파라미터
  - real_data : 실제자료
  - synthetic_data : 재현자료
  - metadata : 싱글테이블 패키지에서 선언한 데이터 (보통 real_data와 일치)
  - column_name : 자신이 표현하고 싶은 열

5. 재현자료의 평가방법

Plot



5. 재현자료의 평가방법

4. Visualization Utilities

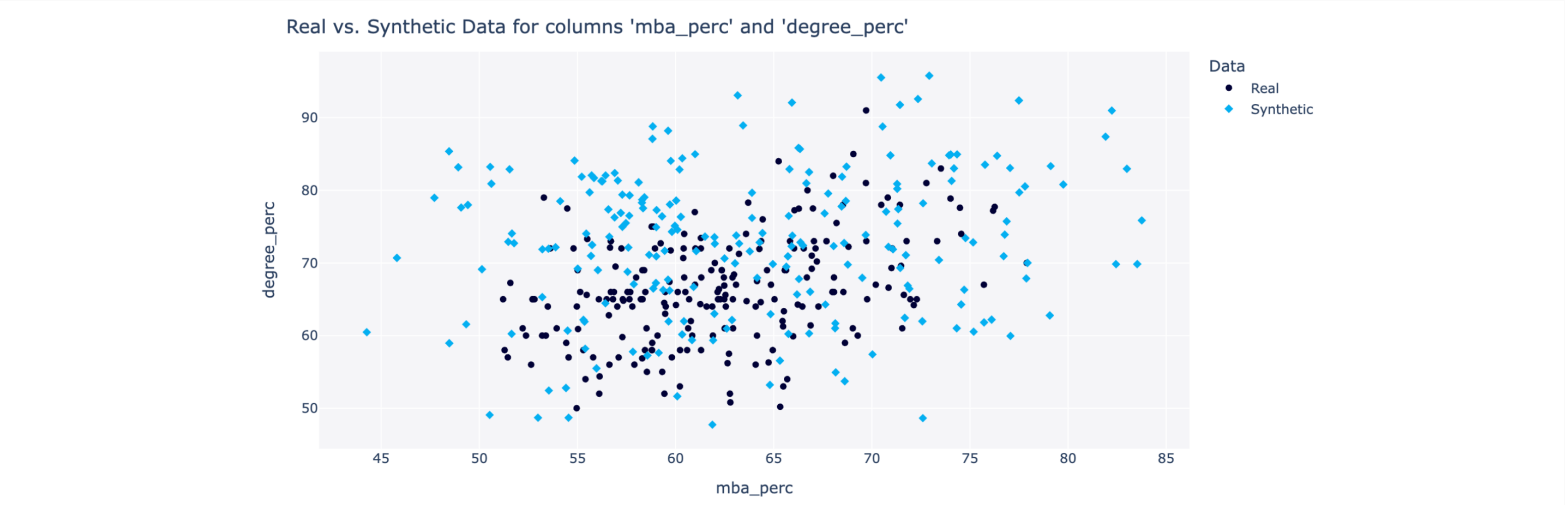
2. 2D형태로 실제자료와 재현자료의 분포 표현

```
from sdmetrics.reports import utils

fig = utils.get_column_pair_plot(
    real_data=real_table,
    synthetic_data=synthetic_table,
    column_names=['mba_perc', 'degree_perc'],
    metadata=my_table_metadata_dict

)

fig.show()
```

- 파라미터
  - real_data : 실제자료에서의 두개의 연속형 Column
  - synthetic_data : 재현자료에서의 두개의 연속형 Column
  - metadata : 싱글테이블 패키지에서 선언한 데이터 (보통 real_data와 일치)
  - column_name : 자신이 표현하고 싶은 열 (2개)

5. 재현자료의 평가방법

Plot



5. 재현자료의 평가방법

5. BoundaryAdherence

측도 BoundaryAdherence는 재현자료의 변수가 실제자료의 최소값과 최대값을 준수하는지의 여부를 측정한다. 실제 경계를 준수하는 재현자료의 행의 백분율을 반환한다.

1. Categorical : 범주형 자료 형태의 열
2. Boolean : True , False 형태의 열

※ 이 메트릭을 사용하려면 두 열이 모두 호환되어야 한다. 열에 결측값이 있으면 메트릭은 이 값을 추가 단일 범주로 처리한다.

점수는 0.0 ∼ 1.0의 형태로 보여주고 1.0으로 갈수록 실제자료와 재현자료의 최소/최대값의 범위를 준수하는 것으로 판단하면 된다.

5. 재현자료의 평가방법

실행 코드

```
from sdmetrics.single_column import BoundaryAdherence

BoundaryAdherence.compute(
    real_data=real_table['column_name'],
    synthetic_data=synthetic_table['column_name']
)
```

- 파라미터
  - real_data : 실제자료에서의 하나의 연속형 Column
  - synthetic_data : 재현자료에서의 하나의 연속형 Column

5. 재현자료의 평가방법

6. CategoricalCoverage

측도 CategoricalCoverage는 재현자료의 열이 실제 데이터의 열에 있는 가능한 모든 범주를 포함하는지 여부를 측정한다.

1. Categorical : 범주형 자료 형태의 열
2. Boolean : True , False 형태의 열

※ 결측값은 무시함.

범위는 0에서 1로 표현할 수 있으며, 1에 가까워 질수록 재현자료의 열에는 실제자료의 열에 있는 모든 고유 범주가 포함됨을 의미한다.

5. 재현자료의 평가방법

실행 코드

```
from sdmetrics.single_column import CategoryCoverage

CategoryCoverage.compute(
    real_data=real_table['column_name'],
    synthetic_data=synthetic_table['column_name']
)
```

- 파라미터
  - real_data : 실제자료에서의 하나의 범주형 Column
  - synthetic_data : 재현자료에서의 하나의 범주형 Column

5. 재현자료의 평가방법

7. MissingValueSimilarity

측도 MissingValueSimilarity는 재현자료에서의 결측값 비율이 실제 데이터의 지정된 열과 동일한지의 여부를 비교한다.

범위는 0에서 1로 표현할 수 있으며, 1에 가까워 질수록 결측값 비율이 동일하다는 것을 의미한다.

이 메트릭은 실제데이터와 재현자료 R ,S 모두 결측값 p의 비율을 계산한다. 정규화 후 [0,1] 범위의 유사도 점수를 반환하며, 1은 가장 높은 유사도를 나타낸다.

5. 재현자료의 평가방법

실행 코드

```
from sdmetrics.single_column import MissingValueSimilarity

MissingValueSimilarity.compute(
    real_data=real_table['column_name'],
    synthetic_data=synthetic_table['column_name']
)
```

- 파라미터
  - real_data : 실제자료에서의 하나의 연속형 Column
  - synthetic_data : 재현자료에서의 하나의 연속형 Column

5. 재현자료의 평가방법

8. NewRowSynthesis

측도 NewRowSynthesis는 재현자료의 각 행이 새롭게 생성된 행인지 또는 실제자료의 원래 행과 정확히 일치하는지 여부(복사본,반복인지 여부)를 측정한다.

1. Categorical : 범주형 자료 형태의 열
2. Boolean : True , False 형태의 열
3. Numerical : 이 메트릭은 연속적인 수치 데이터를 의미한다.
4. Datetime : 이 메트릭은 날짜 시간의 값을 의미한다.

※ 이 메트릭은 결측값에서 일치하는 항목을 찾는다. 실제 데이터에 있을수 있는 다른 열은 무시한다.

범위는 0에서 1이며 1에 가까울수록 실제 데이터와 일치 항목이 없다는 것을 의미한다.

5. 재현자료의 평가방법

실행 코드

```
from sdmetrics.single_table import NewRowSynthesis

NewRowSynthesis.compute(
    real_data=real_table,
    synthetic_data=synthetic_table,
    metadata=single_table_metadata_dict,
    numerical_match_tolerance=0.01,
    synthetic_sample_size=10_000
)
```

- 파라미터
  - real_data : 실제자료에서의 하나의 연속형 Column
  - synthetic_data : 재현자료에서의 하나의 연속형 Column
  - metadata : single_table metadata
  - numerical_match_tolerance : 일치로 간주되기 위해 두 숫자 값이 얼마나 가까워야하는지 나타내는 값(> 0)
  - synthetic_sample_size : 이 메트릭을 계산하기 전에 샘플링 할 재현자료 행의 갯수,

5. 재현자료의 평가방법

9. RangeCoverage

측도 RangeCoverage는 재현자료의 열이 실제 데이터의 열에 있는 모든 범위의 값을 포함하는지 여부를 측정한다.

1. Numerical : 이 메트릭은 연속적인 수치 데이터를 의미한다.
2. Datetime : 이 메트릭은 날짜 시간의 값을 의미한다.

점수가 1이 되면 재현자료의 열은 실제 데이터의 열에 있는 모든 값의 범위를 포함하고 있다고 간주하고

0으로 가까워질수록 실제 데이터의 열과 많이 일치하지 않는다는 것을 의미한다.

5. 재현자료의 평가방법

실행 코드

```
from sdmetrics.single_column import RangeCoverage

RangeCoverage.compute(
    real_data=real_table['column_name'],
    synthetic_data=synthetic_table['column_name']
)
```

- 파라미터
  - real_data : 실제자료에서의 하나의 연속형 Column
  - synthetic_data : 재현자료에서의 하나의 연속형 Column

5. 재현자료의 평가방법

10. StatisticSimilarity

측도 StatisticSimilarity는 요약 통계량을 비교하여 실제자료에서의 열과 재현자료간의 유사성을 측정한다. 지원되는 요약통계량으로는 평균 , 중위수 , 표준편차이다.

1. Numerical : 이 메트릭은 연속적인 수치 데이터를 의미한다.
2. Datetime : 이 메트릭은 날짜 시간의 값을 의미한다.

※ 결측값은 무시함.

0에서 1사이의 범위를 가지며 1로 가까워질수록 재현자료가 실제 자료와 유사하다는 것을 의미한다.

5. 재현자료의 평가방법

실행 코드

```
from sdmetrics.single_column import StatisticSimilarity

StatisticSimilarity.compute(
    real_data=real_table['column_name'],
    synthetic_data=synthetic_table['column_name']
    statistic='mean'
)
```

- 파라미터
  - real_data : 실제자료에서의 하나의 연속형 Column
  - synthetic_data : 재현자료에서의 하나의 연속형 Column
  - statistic : 'mean' , 'median' , 'std'