Random Oversampling

: minor class 예측 시 overfitting 될 가능성이 있다.

SMOTE (Synthetic Minority Oversampling Technique)

소수 샘플을 오버샘플링. **새로운 사례의 데이터 예측**에는 취약할 수 있다. 기존의 데이터 양이 너무 작으면 문제가 될 수 있다.

보통 bootstrapping(중복 허용 randomsampling 이후 평균 구함)이나 KNN 기법(K-Nearest Neighbor)을 이용한다.

SMOTE를 예시를 통해 설명하자면,

신용카드 사기와 같은 데이터의 경우에는

99.82…의 확률로 사기가 아니고,

그 외 0.172749의 확률로 사기가 벌어지기 때문에,

무조건 사기가 아니라고 하는 데이터에도 99.82의 accuracy를 보이고, 높은 Precision을 보이면서 잘 학습되는 것처럼 보일 수 있지만 실제로 중요한 것은 사기를 예측하는 것이기 때문에 Recall값이 높아야 하는데 이런 경우 Recall값이 현저히 떨어지게 되어 목적에 맞는 결과를 내기 어렵다. 이 경우 SMOTE로 Recall 값을 증가시켜주게 되면 정밀도와 F1-score는 현저히 떨어질 수 있으나(Test set의 사기 데이터가 적으므로) Recall 값이 증가하여 목적에 부합하는 효과를 얻을 수 있다.

장점 : overfitting의 확률이 감소하며 정보 손실 우려가 적다.

단점 : minor class의 data를 생성하는 동안 major class data와 겹치거나 침범할 수 있다. 이에 따라 오히려 성능이 감소할 수 있다. 또한 고차원 데이터에 효율적이지 않다.

결론적으로 시각화하여 major class data와 minor class data 가 구분되는 data set에서 Smote를 적용하여 data augmentation하는 것이 적절할 것이다.

MSMOTE(Modified Synthetic Minority Oversampling Technique)

이런 소수 클래스의 **분포**와 잠재적인 noise를 고려하지 않는 SMOTE의 단점을 고려하여 나온 모델이 MSMOTE(Modified Synthetic Minority Oversampling Technique)이다.

알고리즘을 3가지 단계로 나누는데

- security/safe samples
- border samples

- latent(잠재) noise sample로 나눈다

security/safe samples는 classifier의 성능을 높일 수 있는 데이터이며,

noise samples은 성능을 낮추는 데이터이며,

애매한 것은 border samples로 구분된다.

SMOTE가 소수 클래스 데이터에 대해 마구 데이터를 생성했다면 MSMOTE는 security/safe samples 위주로 데이터를 생성하고, Latent noise sample에 대해서는 아무 것도 적용하지 않는다.