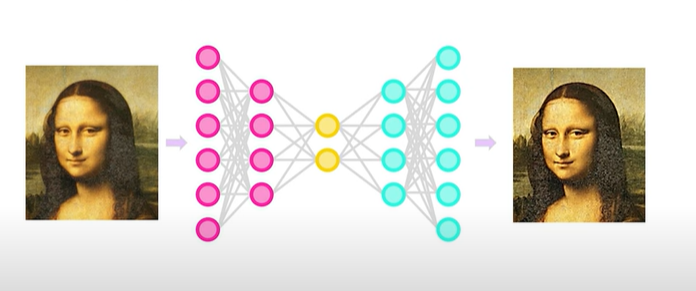
오토인코더(Autoencoder)

셀프지도 학습 방식으로 비지도 학습과 비슷하게 출력값(라벨)이 주어지지 않으며, 주어진 데이터의 특성과 분포를 기반으로 학습한다. 레이블이 없는 데이터만을 사용하여, 원본 데이터와 복원된 데이터 간의 차이인 재구성 손실(reconstruction loss)을 최소화하는 방향으로 진행되며, 이를 통해 주어진 데이터의 중요한 특성을 잡아내게 된다. 쉽게 정리하자면 오토인코더는 입력을 받아서 그 입력과 거의 똑같은 출력을 만들어내는 신경망이다.

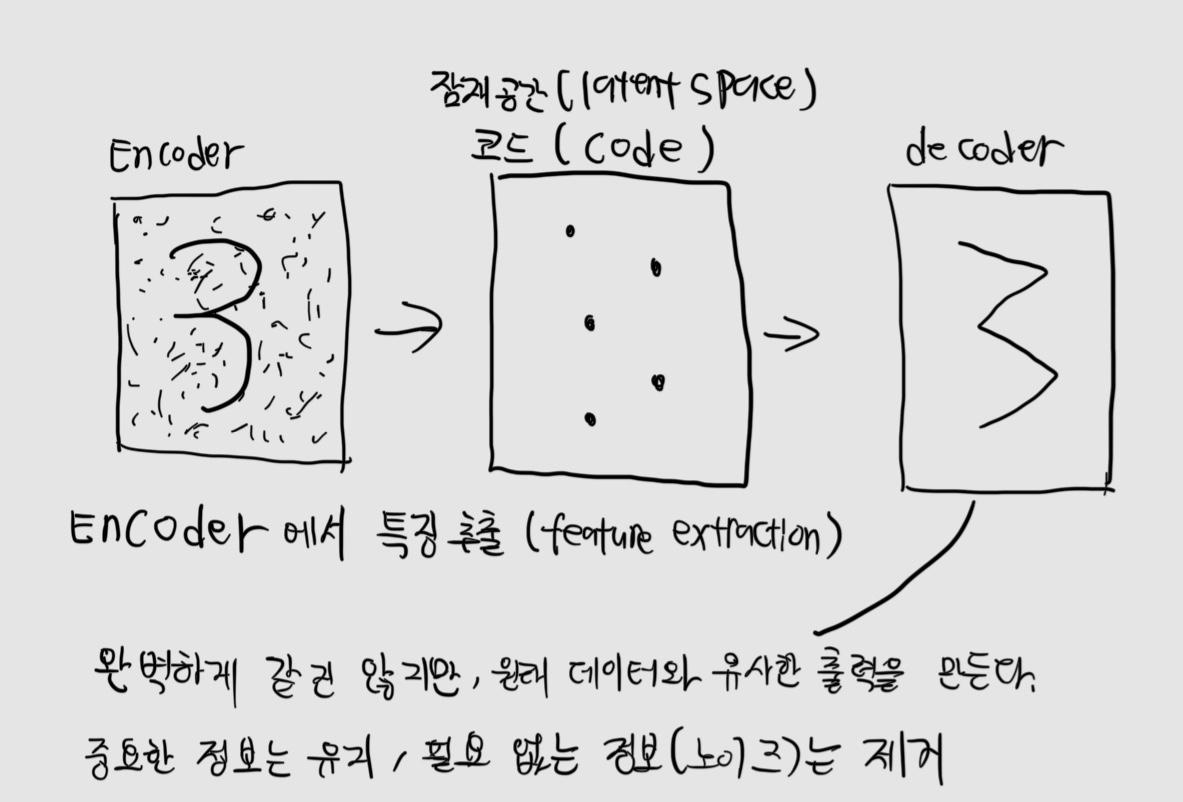


encoder에서 입력 데이터를 받아, 더 작은 차원으로 압축한다. 압축된 것을 우리는 잠재공간(latent space) 또는 코딩(coding)이라고 한다. 그 후 decoder에서 압축된 표현을 다시 원래의 데이터(원본에 가깝게)로 복원하려 한다.

차원 축소는 데이터의 특성을 줄여 계산 효율성을 높이고, 노이즈를 줄이며, 데이터를 더 잘 이해하기 위해 사용된다. 즉 불필요한 정보를 제거하고 중요한 정보를 유지함으로써 모델의 성능을 개선할 수 있다. 많은 양의 데이터를 요약본으로 만들어 핵심 내용만 간추린다고 생각하면 된다. 이 요약본으로 새로운 형태의 원본과 비슷한 데이터를 생성할 수도 있다. 데이터의 중요한 특징은 살리고, 노이즈나 불필요한 정보는 제거하는 방식으로 학습한다.

아래에는 노이즈가 낀 3을 autoencoder방식으로 처리한 과정 예시이다

EX)



오토인코더는 흐릿하거나 손상된 이미지를 깨끗하게 복원하고 흑백 사진을 컬러로 변환할 수 있으며, 저화질 이미지를 고화질로 복원할 수 있다. 우리의 주제인 이상 탐지 시스템에서는 이상한 데이터나 사기 거래, 해킹 시도 등을 찾아 낼 수 있다.

오토인코더는 학습할 떄 정상 데이터만 학습하여 정상 데이터의 특징을 잘 기억하여 정상 데이터는 잘 복원하지만, 한 번도 본 적 없는 이상한 데이터는 못 알아보고 복원이 엉망이 된다. 이상 거래 탐지 시스템에서 정상 거래 데이터를 학습시킨 오토인코더를 만들고, 새로운 거래 데이터를 넣었을 때 복원 오차가 크면 사기일 가능성이 높을 수 있다.

오토인코더는 비정상 데이터를 아예 필요로 하지 않거나, 거의 없이도 이상 탐지를 할 수 있어서 이상거래시스템을 만들 때 유용할 수 있을 것 같다. 복원이 잘 안되는 데이터를 이상치로 잡고, 입력과 출력과정에서 복원 오차가 크면 이상으로 말할 수 있다. 정상/비정상 데이터를 소량 검증셋으로 실험하면서 임계값을 조정하고 복원 오차를 기준으로 임계값을 정하면 된다. Ex) 복원 오차파 0.05 이상이면 “이상으로 판단

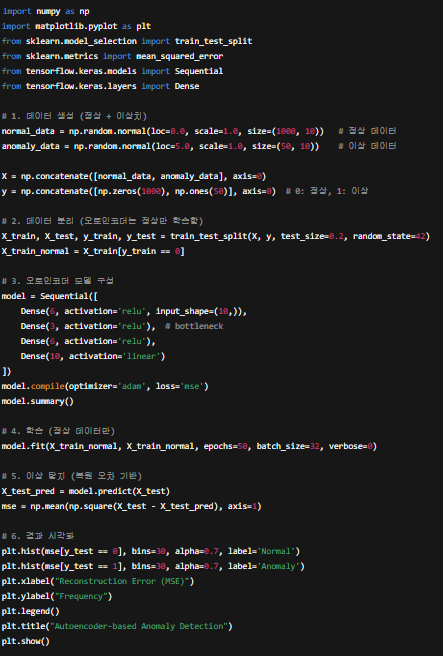
* 정상 거래 패턴만 가지고 오토인코더 학습함
* 새로운 거래가 들어오면 → 오토인코더가 **복원 시도**
* 만약 복원 결과가 원래 입력과 **많이 다르면** → "이건 뭔가 이상하다!"

하지만 오토인코더는 오차가 높은 정상 거래도 존재 할 수 있다 예를 들어 새벽 3시에 5천만 원 거래가 이루어지면 오토인코더는 이상하게 보일 수 있어 오탐 발생 가능성이 있다. 또한 라벨이 없는 비지도 학습 모델이어서 어떤 거래가 진짜 사기인지 알려주는 정답을 몰라 평가나 해석에 어려움이 있을 수 있고, 어디까지가 정상이고, 어디가 이상인지 경계선을 설정하는 기준이 애매하다.

정확도와 설명력을 챙기려면 하이브리드 접근이라고 해서 오토인코더의 재구성 에러와XGBoost를 같이 사용하여 정확도와 설명력을 둘 다 챙길 수 있다. (Autoencoder + XGBoost나 다른 모델 조합)

|  |
| --- |
|  |

아래는 나중에 참고할 오토인코더를 이용한 이상 탐지 코드를 gpt가 작성한 것이다.



XGBOOST

결정트리'를 조합해서 더 똑똑한 예측을 만드는 머신러닝 알고리즘이다. 입력과 정답(레이블)을 함께 학습하는 방식이며 경사하강법을 쓴다. 모델이 틀린 만큼 손실을 계산하고, 그걸 줄이는 방향으로 다음 나무를 만들어 성능을 높여간다. 이 고쳐과는 과정을 반복하여 만든 여러 개의 나무가 함께 작동해서 예측을 하는 게 xgboost이다

결정트리란?

[나이 < 30세] → 예금 안함

[30세 이상이고 연봉 < 5000만원] → 예금 안함

[30세 이상이고 연봉 ≥ 5000만원] → 예금함

이런 식으로 규칙을 만들어서 데이터를 나누어 예측하는 것이 트리이다. 트리를 하나만 쓰면 오버피팅 되거나, 성능이 부족할 수 있다.

boosing이란?

여러 개의 모델(트리)을 순차적으로 만들고, 이전 모델이 틀린 걸 다음 모델이 보완하도록 훈련시키는 방법

첫 번째 트리가 대충 예측 -> 일부는 맞고, 일부는 틀림

두 번째 트리가 첫 번째 트리가 틀린 부분만 집중하여 다시 학습

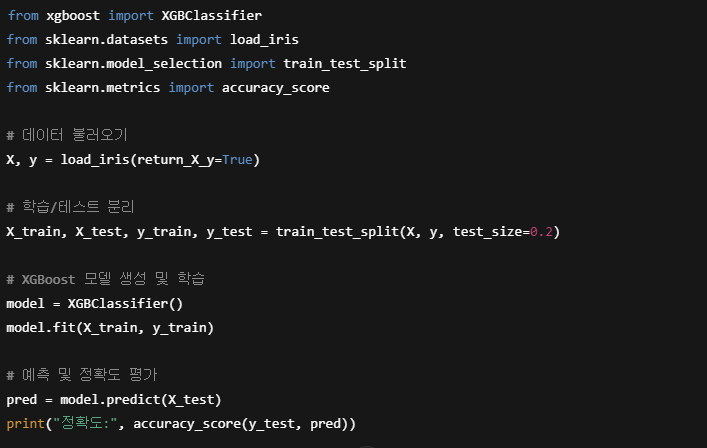
세 번째 트리는 두 번째 트리가 못 잡을 걸 또 보완하고 학습함

이렇게 이어지다가 마지막에 모든 트리들의 결과를 합쳐서 최종 예측을 함

우리 프로젝트에서는 거래 데이터를 넣고, 정상/사기 라벨로 각각 지도학습을 시킨 후, 오차를 줄이는 방향으로 나무를 하나씩 추가하여 점점 더 정확한 사기 탐지기를 만들 수 있다. 우리는 사기 거래가 매우 적어서 xgboost 단독으로 사용하기 보단, 오토인코더와 같은 비지도 학습 기법과 함께 사용해야 성능이 좋아질 것 같다.

만약 왜 이 거래가 사기인지 모델이 사람이 이해할 수 있게 설명해주면 좋겠다 싶으면 xgboost에서는 shap이라는 설명을 도와주는 도구를 써야한다. Shap은 입력값이 모델의 예측에 얼마나 많은 영향을 주었는지 수치로 알려주는 기법이다. 혹은 오토인코더와 같이 사용하게 될 경우, 오토인코더가 계산한 이상도 점수를 사람이 볼 수 있어 shap이 필요없다.

아래는 나중에 참고할 gpt가 쓴 xgboost 코드이다



Autoencoder + XGBoost

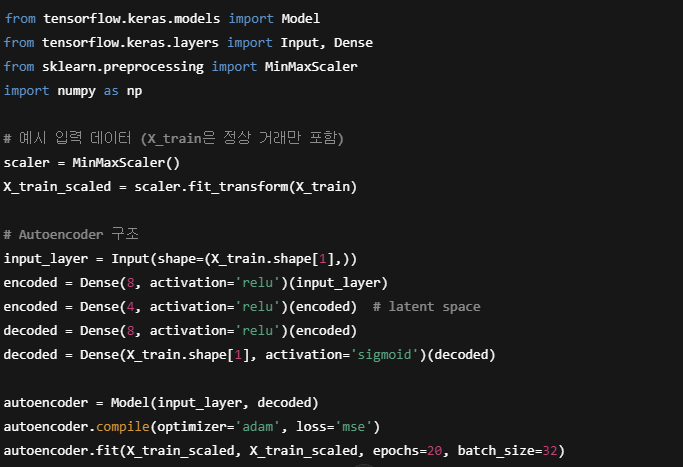
둘을 합치면, Autoencoder로 이상도 점수를 추출하고 XGBoost가 이를 바탕으로 예측하고 설명까지 가능하다.

구조

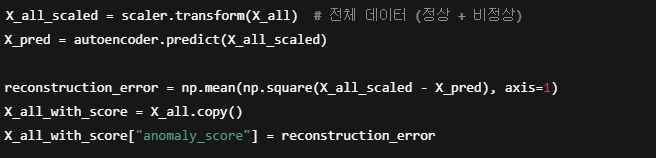
거래 데이터 (입력) -> Autoencoder (정상 패턴 학습) -> 재구성 에러 (이상도 점수) -> XGBoost (에러 + 원본 데이터 기반으로 사기 여부 예측) -> SHAP (왜 사기인지 설명)

이러한 구조로 프로젝트를 한다면 나중에 참고하기 위해 gpt로 코드를 작성해 보았다.

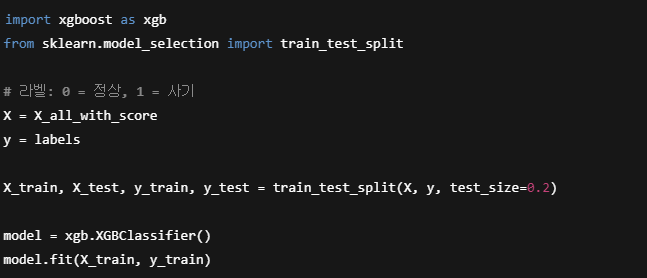
Autoencoder 훈련 (정상 거래만)



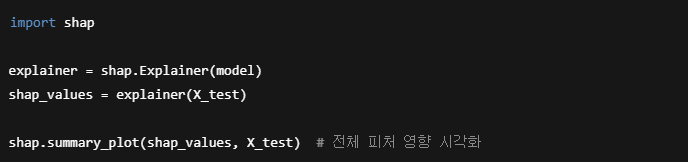
이상도 점수 계산 (정상 + 사기 데이터)



XGBoost로 사기 예측 학습



SHAP으로 설명



실습 코드는 깃허브에 참고용으로 올려보겠습니다.