딥러닝 HW1(17.11.02)

제출자 : 김 신

1. 사용한 모델 혹은 알고리즘

- 사용한 모델 : Logistic Regression, **Random Forest**
- 모델 설명
  1) Logistic Regression
     : Logestic Regression은 우리가 찾고자하는 분류 값(종속변수)이 0 혹은 1 이렇게 2개로 나뉘어있고, 1개 이상의 X(독립변수)가 있을 때 이를 분석하기 위해 사용하는 모델이다.
  2) Random Forest
     : Decision Tree의 주요 단점인 과대적합을 회피할 수 있는 방법이다. Random Forest는 조금씩 다른 여러 결정 트리의 묶음인데, 이 트리들이 동일한 레코드마다 분석한 분류 값을 서로 비교하여 우세한 값을 선택하는 모델이다. 트리의 개수를 나타내는 인자는 n_estimator인데 n_estimator는 클수록 좋다. 하지만 n_estimator 값이 너무 높으면 연산하는 시간이 매우 오래 걸리는 단점이 있다.

2. 인자 탐색 과정 및 결과

- 결정해야할 HyperParameter 및 탐색 과정
  1) max_depth / n_estimator
     : **GridSearchCV로 탐색**할 때 max_depth는 range(2,10)로 주었고 n_estimator은 [2000, 1000, 500, 100]로 주어 이 값들 중에서 최적의 값을 찾음.
  2) 학습에 사용할 변수
     : **RandomForest 모델에서 .feature_importance Attribute를 사용**하여 중요한 피쳐들을 살펴보았음

- 결과
  1) max_depth / n_estimator
     : max_depth는 6, n_estimator는 2000으로 결정되었음.
  2) 학습에 사용할 변수
     : 중요도가 1% 미만인 변수들은 제외시킴. 그 결과 11개의 변수가 선택됨.
     ※ 1% 미만인 변수들을 선택하면 10개의 변수가 선택되어야 하지만, 10개부터 20개까지 변수를 선택하여 돌려보았는데, 11개일 때 가장 높은 F값이 나와 11개로 선택하였습니다.

3. 데이터 조작 과정 및 결과

- 최초에 시도한 방법은 비정상 거래가 292개로 비중이 매우 작기 때문에 정상데이터를 292개 뽑아서 모델을 만들어서 분석해보았습니다. 분석결과 F값은 약 0.83 그렇게 나쁘진 않았지만, 주어진 데이터에 비해서 너무 작은 양의 데이터를 사용해서 그런지 random state값을 수정하면 F값 변동 폭이 매우 커 신뢰할 만한 모델이라 볼 수 없었습니다. 이때는 베타 테스트라 생각하고 인자들은 특별히 손대지 않고 logistic regression과 Random Forest 모델로 분석해보았습니다.

- 최초에서 시도한 방법에서 찾아낸 문제점들은 다음과 같다
  1) 학습에 사용할 데이터를 너무 작게 선택하였음
  2) 학습에 큰 영향을 줄 수 있는 인자들을 기본 값들로 설정함.

- 위의 문제점들을 다음과 같이 개선함.
  1) 데이터들을 모두 사용하고 class weight를 balanced로 주어 데이터 불균형을 해소함.
  2) GridSearchCV 및 RandomForest의 .feature_importance Attribute를 사용하여 주요 인자들과 피쳐 개수에 대한 최적의 값을 찾아냄

- 개선사항을 반영한 모델로 Validation을 분석한 결과

Precision	약 0.95
Recall	약 0.81
F-Score	약 0.876