서울대학교 4 차 산업혁명 아카데미

빅데이터 플랫폼

딥러닝 (강유 교수님) 숙제 1

출제: 2017 년 10 월 26 일 목요일

제출: 2017 년 11 월 02 일 목요일

본 숙제의 목표는 실제 데이터에서 잘 동작하는 기계학습 모델을 만들어 보는 것입니다. 사용 언어는 Python 3 이며 scikit-learn 라이브러리에 구현된 기계학습 모델을 자유롭게 사용할 수 있습니다. 숙제에 사용할 데이터와 뼈대 코드가 제공됩니다.

1. 데이터 정보

숙제에 사용되는 데이터는 총 49,200 개의 신용 카드 거래 내역입니다. 그중 1%인 492 개의 거래는 "비정상적인" 거래이며 나머지 48,708 개의 거래는 "정상적인" 거래입니다. 여기서 비정상적인 거래는 도난이나 사기 등에 의해 발생된 거래를 의미합니다. 개개의 거래는 총 30 개의 속성과 1 개의 클래스를 포함하며, 이는 다음과 같습니다.

- **시각(Time):** 각 거래가 발생한 상대적인 시각을 나타냅니다. 전체 데이터에서 가장 먼저 일어난 거래의 시각은 0 이며 모든 값은 음이 아닌 정수입니다.
- **임의 속성(V1-28):** 개인 정보 때문에 자세한 내용을 밝힐 수 없는 개별 거래 속성입니다. 총 28 개의 속성이 있으며 앞의 속성일수록 뒤의 속성보다 높은 분산을 가지고 있습니다. 즉 V1 속성이 V28 속성에 비해 높은 분산을 가지고 있습니다. 그러나 높은 분산이 높은 예측력을 의미하는 것은 아니므로 사용할 속성을 신중히 결정해야 합니다.
- **거래량(Amount):** 개별 거래에서 주고 받은 돈의 양을 의미합니다.
- **클래스(Class):** 개별 거래가 정상적인 거래인지 혹은 비정상적인 거래인지 나타냅니다. 0 이면 정상적인 거래, 1 이면 비정상적인 거래를 의미합니다.

2. 문제 정의 및 평가 방법

본 숙제의 목표는 "비정상" 거래를 최대한 잘 찾아내는 기계학습 모델을 만드는 것입니다. 주어진 데이터의 클래스 분포가 매우 치우쳐 있으므로 모델을 설계할 때 이 점을 잘 고려해야 합니다. 결과는 다음과 같이 정의되는 F1 점수를 통해 평가합니다.

$$F1Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

여기서 정밀도(precision)는 모델이 비정상이라고 예측한 거래 중 실제 비정상 거래의 비율을 의미합니다. 즉 100 개의 거래를 비정상이라고 예측했는데 그중 60 개가 실제로 비정상이었다면 정밀도는 60%가 됩니다. 재현율(recall)은 전체 비정상 거래 중 모델이 성공적으로 예측한 거래의 비율을 의미합니다. 전체 비정상 거래가 100 개인데 그중 60 개를 예측하는 데 성공했다면 재현율은 40%가 됩니다. 예를 들어 모든 거래를 비정상이라고 예측한다면 정밀도는 1%, 재현율은 100%가 됩니다. 모든 거래를 정상이라고 예측한다면 (예측 시도한 거래가 없으므로) 정밀도가 올바르게 정의되지 않으나, 본 숙제에서는 0%라고 가정하겠습니다. F1 점수는 정밀도와 재현율을 모두 고려하는 평가 방식이며 가장 널리 쓰이는 분류 모델 평가 방식 중 하나입니다. 정밀도와 재현율이 모두 높아야만 높은 F1 점수를 얻을 수 있습니다.

3. 제공 데이터

본 숙제에는 훈련, 검증, 테스트에 사용할 총 세 개의 데이터 파일이 제공됩니다. 테스트 데이터에는 클래스 정보가 포함되어 있지 않으므로 모델 훈련에 사용할 수 없습니다. 본 숙제의 목표는 테스트 데이터에서 잘 동작하는 모델을 만드는 것이기 때문에 훈련 데이터와 검증 데이터를 사용하는 방식은 자유이며, 서로의 비율이나 역할을 변경해도 상관 없습니다.

- **훈련 데이터(train.csv):** 292 개의 비정상 거래와 28,908 개의 정상 거래가 포함된 훈련 데이터입니다. 모델을 훈련하는 데 사용하기를 권장합니다.
- **검증 데이터(valid.csv):** 100 개의 비정상 거래와 9,900 개의 정상 거래가 포함된 검증 데이터입니다. 모델 성능을 검증하고 인자를 탐색하는 데 사용하기를 권장합니다.
- **테스트 데이터(test.csv):** 100 개의 비정상 거래와 9,900 개의 정상 거래가 포함된 테스트 데이터입니다. 제공되는 데이터에는 클래스 정보가 포함되어 있지 않습니다. 즉 이 데이터는 모델을 학습하는 과정에서 전혀 사용할 수 없습니다.

4. 코드 구현

여러분의 코드(main.py)는 테스트 데이터(test.csv)를 읽어서 예측한 각 거래의 클래스를 파일로 출력해야 합니다. 뼈대 코드가 제공되며 여러분은 predict 함수의 내용을 채워 제출해야 합니다. 라이브러리를 사용하는 것은 자유이며, 다음과 같은 라이브러리를 권장합니다.

- **scikit-learn:** 다양한 기계학습 모델을 구현한 라이브러리입니다. 본 숙제에서는 필수적으로 사용해야 합니다.
- **pandas:** 테이블 형태로 파일을 읽거나 쓸 수 있는 라이브러리입니다. 배우는 데 다소 노력이 필요하지만 익숙해질 경우 효율적으로 데이터를 관리할 수 있습니다.
- **numpy:** 벡터와 행렬 형태의 데이터를 조작하는 데 특화된 라이브러리입니다. tensorflow 등 많은 라이브러리가 본 라이브러리를 기반으로 구현되어 있습니다.

5. 보고서 작성

여러분은 1-2 페이지의 간단한 보고서를 작성하여 제출해야 합니다. 본 숙제는 모델의 객관적인 성능 뿐 아니라 여러분이 좋은 모델을 찾기 위해 노력한 과정을 기반으로 채점됩니다. 따라서 여러분의 보고서는 다음과 같은 내용을 필수적으로 포함해야 합니다.

- **사용한 모델 혹은 알고리즘:** 사용한 모델을 깊이 이해할 필요는 없으나 각 모델의 핵심적인 아이디어 정도는 파악하고 있어야 합니다.
- **인자 탐색 과정 및 결과:** 여러분이 사용한 모델의 최적 인자를 탐색하는 과정, 방법, 그로 인해 찾은 최적 인자를 서술해야 합니다.
- **데이터 조작 과정 및 결과:** 여러분은 주어진 데이터를 그대로 사용할 필요가 없습니다. 속성들을 합치고, 각 속성의 분포를 조작하고, 꼭 필요한 속성을 고르는 등 데이터를 조작하는 과정은 좋은 모델을 만드는 데 필수적입니다.

6. 제출 방법

완성된 숙제는 압축하여 유재민 조교(jaeminyoo@snu.ac.kr)에게 보내면 됩니다. 압축 파일의 이름에는 제출자의 이름이 반드시 포함되어 있어야 합니다. 압축 파일에 포함되어야 하는 파일의 목록은 다음과 같습니다. 이중 코드 파일(main.py)은 Python 3 환경에서 실행 가능해야 하며, 데이터 파일(test.csv)이 주어졌을 때 결과 파일(result.csv)을 만들어야 합니다.

- HW1_{이름}.zip
    - **report.pdf:** 보고서 파일입니다. PDF 형식이어야 합니다.
    - **main.py:** 결과 코드입니다.
    - **result.csv:** 여러분의 코드를 실행한 예측 결과입니다.
    - **README.txt:** (선택사항) 추가적으로 언급할 내용을 적으면 됩니다.