|  |  |  |  |
| --- | --- | --- | --- |
| **2023-1 머신러닝 및 실습** | | | |
| 학번 | 2021136150 | 이름 | 양동재 |

이번 실습시간에서는 Data Evaluation Methods를 학습함으로써, 모델의 성능을 평가하는 다양한 이론에 대해 이해하고, 이를 적용하여 인공지능 모델에 대한 문제 정의를 명확히 하고 문제를 잘 해결했는지 평가할 수 있는 Evaluation method를 선택하는 것이 주 목적입니다. AI class-based data에 대한 편향이 존재할 때 잘못된 Model evaluation method를 채택하면, 모델이 실제 세계에 적용되었을 때 심각한 성능 저하를 발생시킬 수 있습니다. 따라서 본 교과과정에서 학습한 다양한 Model evaluation method를 활용하여 문제에 적합한 평가 방법으로 학습된 모델을 평가해야 합니다.

학습된 모델을 실제 세계 데이터에 높은 성능으로 적용시키기 위해 다양한 기법들이 활용되고 있습니다. 대표적으로 테스트 데이터가 부족한 경우를 고려하여 학습용 데이터를 n분할하여 성능 평가를 하는 방법인 cross validation, 데이터 편향 문제를 해결하는데 기반이 되는 confusion matrix와 micro, macro averaging method가 있습니다. 각 Model evaluation method 간의 특징을 파악하고 활용하는 것은 모델이 실제 세계 데이터에 적용되었을 때 높은 성능을 보여주는 중요한 요소입니다.

따라서, 이번 실습에서는 다양한 데이터와 학습 환경에서 다양한 모델 평가방법을 적용하여 모델 평가 방법론에 대해 이해를 하고, 어떤 데이터 특성 및 학습 환경에서 어떤 모델 평가 방법을 선택하는 것이 적합한지에 대한 이해를 높이는 것이 목표입니다.

**[데이터]**

Shape, arrow

Description automatically generatedMNIST 데이터셋은 70,000개의 손글씨 숫자 이미지 데이터셋으로, 각 이미지는 해당 숫자와 함께 라벨링되어 있으며, 이미지 분류 작업의 벤치마크 데이터셋으로 광범위하게 활용되고 있습니다. 데이터셋의 이미지는 28x28 크기의 그레이스케일로, 중앙에 위치하며, 이 데이터셋은 60,000개의 훈련 이미지와 10,000개의 테스트 이미지로 구성되어 있습니다. MNIST 데이터셋은 머신러닝 분야에서 많은 알고리즘, 특히 로지스틱 회귀, 신경망, 서포트 벡터 머신 등의 성능 평가에 사용됩니다. 이러한 알고리즘은 MNIST 데이터셋을 통해 이미지 분류와 같은 다양한 문제를 해결하는 데 활용됩니다.

**[문제1]**

1. **문제정의**

이번 실습 문제의 목표는 임의의 행렬 값을 사용하여 confusion matrix를 구현하고, confusion matrix의 요소들이 의미하는 바를 명확히 이해하며, confusion matrix 요소 간의 연산을 통해 다양한 모델 평가 방법을 개발하는 것입니다.

이를 위해서는 먼저 분류 문제에서 모델이 어떤 과정을 통해 학습하고, 확률을 계산하여 예측하는지에 대해 이해해야 합니다.

실습에서는 머신러닝 전체 주기 중 모델 평가 과정에서 어떤 평가 방법론을 적용할지 분석하게 되며, 이를 통해 적절한 모델 평가 방법론을 선택하는 방법에 대해 학습하게 됩니다.

결론적으로, 이번 실습 문제는 전반적인 분류 문제의 머신러닝 과정을 이해하고, 다양한 모델 평가 방법론을 적용하여 각각의 방법에 대해 이해하는 것이 목표입니다.

1. **학습전략**

**A. Hypothesis**

- Confusion matrix의 개념과 요소를 이해한다.

- Accuracy, Precision, Recall, F1 score의 개념과 차이점을 이해한다.

- Confusion matrix를 활용하여 Accuracy, Precision, Recall, F1 score를 계산하는 공식을 이해한다.

- Confusion matrix를 입력 받아 Accuracy, Precision, Recall, F1 score를 계산하는 함수를 구현한다.

- 구현한 함수를 적절한 테스트 데이터셋에 적용하여 결과를 확인한다.

**B. EDA**

본 데이터셋의 특징은 60,000장의 손글씨 이미지 데이터로 28\*28의 해상도이며 총 784개의 feature를 갖고있습니다.

1. **자신이 구현한 함수 및 클래스에 대한 설명 또는 유도 과정 작성**

이 문제를 해결하기 위해 아래와 같은 함수를 구현하였습니다.

1. confusion\_matrix(predictions, labels)

- predictions : 모델이 예측한 결과를 담은 리스트

- labels : 실제 레이블 정보를 담은 리스트

- return : predictions과 labels을 기반으로한 confusion matrix 반환

- 함수 내부에서는 2x2의 빈 행렬을 생성하고, predictions와 labels을 이용하여 각 셀을 채워나갑니다.

- 반환값은 2x2의 confusion matrix입니다.

2. calculate\_metrics(confusion\_matrix)

- confusion\_matrix : confusion matrix 정보를 담고 있는 리스트

- return : accuracy, precision, recall, f1 score 정보를 담은 딕셔너리

- 함수 내부에서는 confusion matrix의 TP(True Positive), TN(True Negative), FP(False Positive), FN(False Negative) 값을 계산하고, 이를 이용하여 accuracy, precision, recall, f1 score를 계산합니다.

- 반환값은 accuracy, precision, recall, f1 score 정보를 담은 딕셔너리입니다.

위 함수들은 각각 confusion matrix를 구하고, 이를 기반으로한 다양한 metrics를 계산하는 함수입니다. 이 함수들은 다음과 같은 과정을 통해 유도하였습니다.

1. confusion matrix

- 모델의 예측 결과와 실제 레이블 정보를 기반으로 confusion matrix를 구하는 것이 필요합니다.

- 이를 위해서는 predictions와 labels 리스트를 입력으로 받아, 이를 기반으로한 confusion matrix를 반환하는 함수를 구현해야 합니다.

- 모델의 예측 결과와 실제 레이블 정보는 각각 True, False로 이루어진 이진 분류 문제를 가정하고 있으며, 따라서 confusion matrix는 2x2 행렬로 구성됩니다.

2. accuracy

- accuracy는 모델이 정확하게 예측한 비율을 나타내는 지표입니다.

- 이를 계산하기 위해서는 confusion matrix에서 TP(True Positive), TN(True Negative), FP(False Positive), FN(False Negative) 값을 이용하여 accuracy를 계산하는 공식을 적용해야 합니다.

3. precision

- precision은 모델이 Positive라고 예측한 것 중에서 실제로 Positive인 비율을 나타내는 지표입니다.

- 이를 계산하기 위해서는 confusion matrix에서 TP(True Positive), FP(False Positive) 값을 이용하여 precision을 계산하는 공식을 적용해야 합니다.

4. recall

- recall은 실제 Positive 중에서 모델이 Positive라고 예측한 비율을 나타내는 지표입니다.

- 이를 계산하기 위해서는 confusion matrix에서 TP(True Positive), FN(False Negative) 값을 이용하여 recall을 계산하는 공식을 적용해야 합니다.

5. f1 score

- f1 score는 precision과 recall의 조화

|  |  |
| --- | --- |
| Confusion Matrix Evaluation Method | Formula |
| Accuracy |  |
| Precision |  |
| Recall |  |
| F1 score |  |

**4. 결과 비교 및 분석**

|  |  |
| --- | --- |
| Confusion Matrix Evaluation Method | Score |
| ACC | 0.45 |
| Precision | 0.46 |
| Recall | 0.60 |
| F1\_score | 0.52 |

위 결과를 보면, class 1에 대해 모델이 예측한 결과(pred1)와 실제 레이블(label1) 정보를 바탕으로 confusion matrix가 계산되었습니다. Confusion matrix를 통해, 예측 결과의 TP(True Positive), TN(True Negative), FP(False Positive), FN(False Negative) 값을 파악할 수 있습니다.

위 결과에서는, class 1에 대한 예측 결과에서는 6개의 TP, 7개의 FN, 4개의 FP, 그리고 3개의 TN이 존재합니다. 이를 바탕으로 Accuracy, Precision, Recall, F1 Score를 계산하였고, Accuracy는 0.45, Precision은 0.46, Recall은 0.60, 그리고 F1 Score는 0.52가 나왔습니다.

Accuracy는 전체 데이터 중에서 모델이 예측한 결과와 실제 레이블 정보가 일치하는 비율을 나타내며, 이 값이 낮다는 것은 모델이 예측을 잘못하고 있다는 것을 의미합니다. Precision은 모델이 Positive로 예측한 것 중에서 실제로 Positive인 비율을 나타내며, Recall은 실제 Positive 중에서 모델이 Positive라고 예측한 비율을 나타냅니다. F1 Score는 Precision과 Recall의 조화 평균으로, Precision과 Recall이 모두 높은 경우에 높은 값을 가지게 됩니다.

위 결과를 바탕으로, 모델의 성능이 좋지 않은 것으로 판단됩니다. Precision 값이 Recall 값보다 낮게 나왔는데, 이는 모델이 Positive로 잘못 예측하는 경우가 많다는 것을 의미합니다. 따라서 모델의 개선이 필요하다고 판단됩니다.

**[문제2]**

1. **문제정의**

이번 실습에서는, 서로 다른 두 클래스에 대한 성능의 평균을 구하는 Micro & Macro averaged precision 함수를 구현하는 것이 목표입니다. 이를 위해서는 먼저 confusion matrix를 계산하는 함수가 필요합니다. Confusion matrix는 모델의 예측 결과와 실제 레이블 정보를 비교하여 TP(True Positive), TN(True Negative), FP(False Positive), FN(False Negative) 값을 계산한 것입니다. 이를 바탕으로 Precision, Recall, F1 Score 등의 성능 지표를 계산할 수 있습니다.

Micro & Macro averaged precision은 이진 분류 문제에서 많이 사용되는 성능 지표 중 하나로, Precision 값을 계산한 후, 두 클래스의 Precision 값을 평균한 값을 반환합니다. Micro averaging과 macro averaging의 차이는, Micro averaging은 각 클래스별 TP, TN, FP, FN 값을 모두 더한 후, 전체 TP, TN, FP, FN 값을 계산하여 성능 지표를 계산하는 방법이며, Macro averaging은 각 클래스의 성능 지표를 계산한 후, 평균 값을 계산하여 성능 지표를 계산하는 방법입니다.

따라서, 이번 실습에서는 다음과 같은 과정을 거쳐 Micro & Macro averaged precision 함수를 구현할 것입니다.

1. confusion matrix 함수 구현

- 모델의 예측 결과와 실제 레이블 정보를 비교하여 confusion matrix를 계산하는 함수를 구현합니다.

- 이를 바탕으로 Precision 값을 계산할 수 있습니다.

2. Micro averaged precision 함수 구현

- confusion matrix에서 TP, TN, FP, FN 값을 모두 더한 후, Precision 값을 계산합니다.

- 이를 이용하여 두 클래스의 Precision 값을 평균한 값을 반환합니다.

3. Macro averaged precision 함수 구현

- 각 클래스의 Precision 값을 계산한 후, 이를 평균한 값을 반환합니다.

위 과정을 통해 Micro & Macro averaged precision 함수를 구현할 수 있습니다.

1. **학습전략**

**A. Hypothesis**

이번 문제를 해결하기 위한 학습 전략은 다음과 같습니다.

1. Micro & Macro averaged precision의 개념과 차이점을 이해한다.

- Micro & Macro averaged precision은 두 클래스에 대한 Precision 값을 평균한 값을 반환하는 지표입니다.

- Micro averaging과 macro averaging의 차이점을 이해해야 합니다.

2. confusion matrix를 활용하여 Precision 값을 계산하는 공식을 이해한다.

- Precision은 TP(True Positive) 값과 FP(False Positive) 값을 이용하여 계산할 수 있습니다.

3. 두 개의 클래스에 대한 confusion matrix를 입력으로 받아 Micro & Macro averaged precision을 반환하는 함수를 구현한다.

- Micro averaged precision은 confusion matrix에서 TP, TN, FP, FN 값을 모두 더한 후, Precision 값을 계산하여 두 클래스의 Precision 값을 평균한 값을 반환합니다.

- Macro averaged precision은 각 클래스별로 Precision 값을 계산하여, 이를 평균한 값을 반환합니다.

4. 구현한 함수를 테스트 데이터셋에 적용하여 결과를 확인한다.

위 전략을 바탕으로, Micro & Macro averaged precision 함수를 구현하고 테스트 데이터셋에 적용하여 결과를 확인합니다. 이를 통해, 두 클래스에 대한 모델의 성능을 측정하고 개선할 수 있습니다.

**B. EDA**

본 데이터셋의 특징은 60,000장의 손글씨 이미지 데이터로 28\*28의 해상도이며 총 784개의 feature를 갖고있습니다.

1. **자신이 구현한 함수 및 클래스에 대한 설명 또는 유도 과정 작성**

1. Confusion Matrix

confusion\_mat 함수는 모델의 예측 결과와 실제 레이블 정보를 비교하여 confusion matrix를 계산하는 함수입니다. 모델이 예측한 레이블 정보(pred)와 실제 레이블 정보(label)를 입력으로 받으며, 반환값으로 confusion matrix를 출력합니다.

2. Micro & Macro averaged precision

precision 함수는 confusion matrix를 입력으로 받아 Precision 값을 계산하는 함수입니다. Micro\_averaged\_precision 함수는 confusion matrix 리스트를 입력으로 받아, Micro averaged precision 값을 반환합니다. 각 confusion matrix에서 TP(True Positive)와 FP(False Positive) 값을 더한 후, 이를 이용하여 Precision 값을 계산하고 두 클래스에 대한 Precision 값의 평균을 반환합니다. Macro\_averaged\_precision 함수는 confusion matrix 리스트를 입력으로 받아, 각 클래스별 Precision 값을 계산한 후, 이를 평균하여 Macro averaged precision 값을 반환합니다.

**4. 결과 비교 및 분석**

위 코드의 결과를 보면, Micro-averaged precision은 0.53, Macro-averaged precision은 0.61로 나왔습니다. 두 지표를 비교해보면, Macro-averaged precision이 더 높은 것을 확인할 수 있습니다.

이는, Micro-averaged precision이 두 클래스의 샘플 수가 차이가 크면, 샘플 수가 많은 클래스의 Precision 값이 높게 나온다는 문제점이 있기 때문입니다. 반면에 Macro-averaged precision은 각 클래스의 Precision 값의 평균을 내기 때문에 샘플 수가 많은 클래스의 Precision 값이 크게 반영되지 않습니다.

따라서, 클래스별로 Precision 값을 따로 계산하여 평균을 구하는 Macro-averaged precision 지표를 사용하는 것이 더 적절할 수 있습니다. 위 결과에서도 Macro-averaged precision이 더 높게 나온 것을 확인할 수 있으므로, 이 지표를 사용하는 것이 더 적절합니다.

**[Extra 문제1]**

1. **문제정의**

이번 문제에서는, mean Average Precision(mAP)를 계산하는 함수를 구현하는 것이 목표입니다. mAP는 Object Detection 등의 컴퓨터 비전 분야에서 모델의 성능을 측정하는 지표 중 하나입니다. mAP를 계산하기 위해서는 Precision-Recall curve와 confidence score, label, threshold 정보가 필요합니다.

Precision-Recall curve는 모델의 성능을 측정하는 데에 자주 사용되는 그래프입니다. Precision-Recall curve는 Precision과 Recall 값을 이용하여 그려지는 곡선입니다. Precision-Recall curve를 그리기 위해서는 모델의 예측 결과와 실제 레이블 정보를 이용하여 Precision과 Recall 값을 계산해야 합니다.

또한, mAP를 계산하기 위해서는 각 클래스별로 Precision-Recall curve를 계산한 후, 이를 평균하여 mAP 값을 구합니다. 이를 위해서는 confidence score, label, threshold 정보가 필요합니다. confidence score는 모델이 해당 클래스에 대해 예측한 확률값입니다. label은 해당 클래스의 실제 레이블 정보를 나타냅니다. threshold는 모델이 해당 클래스로 분류하기 위한 최소한의 확률값을 나타냅니다.

1. **학습전략**

**A. Hypothesis**

1. Precision-Recall curve 계산 함수 구현

- 모델의 예측 결과와 실제 레이블 정보, threshold 값을 이용하여 Precision-Recall curve를 계산하는 함수를 구현합니다.

2. mAP 계산 함수 구현

- 각 클래스별로 Precision-Recall curve를 계산한 후, 이를 평균하여 mAP 값을 구하는 함수를 구현합니다.

3. 구현한 함수를 테스트 데이터셋에 적용하여 결과를 확인합니다.

위 과정을 바탕으로, mAP를 계산하는 함수를 구현하고 테스트 데이터셋에 적용하여 결과를 확인합니다. 이를 통해, 모델의 성능을 측정하고 개선할 수 있습니다.

**B. EDA**

본 데이터셋의 특징은 60,000장의 손글씨 이미지 데이터로 28\*28의 해상도이며 총 784개의 feature를 갖고있습니다.

1. **자신이 구현한 함수 및 클래스에 대한 설명 또는 유도 과정 작성**

위 코드에서는, Precision-Recall curve와 mAP를 계산하는 함수를 구현하였습니다.

`compute\_pr\_curve` 함수는 입력으로 모델이 예측한 confidence score와 실제 레이블 정보, 분류 기준 threshold 값을 받습니다. 먼저, score 값을 내림차순으로 정렬하고, 정렬된 인덱스를 구합니다. 그 후, 정렬된 scores와 labels 값을 이용하여 Precision-Recall curve를 계산합니다. 이를 위해, 전체 양성 샘플 수를 구하고, 이를 이용하여 Precision과 Recall 값을 계산합니다. 마지막으로, threshold 이상인 값 중 가장 작은 인덱스를 찾아, 이를 이용하여 Precision과 Recall 값을 계산합니다.

`compute\_map` 함수는 입력으로 모델이 예측한 confidence score와 실제 레이블 정보를 받습니다. 이 함수에서는 각 클래스별로 Precision-Recall curve를 계산하고, 각 클래스별로 AP 값을 계산합니다. AP 값은 Precision-Recall curve 아래 영역의 면적을 의미합니다. 각 클래스별로 AP 값을 계산한 후, 이를 평균하여 mAP 값을 계산합니다.

이렇게 구현된 함수를 이용하면, Object Detection 등의 컴퓨터 비전 분야에서 모델의 성능을 측정할 수 있습니다. 모델의 성능이 좋아질수록, Precision-Recall curve의 면적이 커지고, mAP 값도 높아집니다.

**4. 결과 비교 및 분석**

Chart, line chart

Description automatically generated

위 코드의 결과는 Precision-Recall curve와 mAP 값을 계산한 결과입니다.

Precision-Recall curve는 해당 모델이 예측한 결과와 실제 레이블 정보를 이용하여 그려진 곡선입니다. 이 곡선은 Precision과 Recall 값을 이용하여 계산됩니다.

mAP 값은 Precision-Recall curve 아래 영역의 면적을 의미합니다. 이 값은 각 클래스별로 Precision-Recall curve를 계산하고, 이를 평균하여 계산됩니다. mAP 값이 높을수록 모델의 성능이 좋은 것을 의미합니다.

위 코드에서는 테스트 데이터셋에 대한 Precision-Recall curve와 mAP 값을 계산하였습니다. 이를 통해, 해당 모델의 성능을 측정하고 개선할 수 있습니다. 만약, Precision-Recall curve의 면적이 작고, mAP 값이 낮다면 모델의 성능이 좋지 않다는 것을 의미합니다. 따라서, 이를 개선하기 위해 다양한 실험을 수행하고, 모델을 개선해 나갈 필요가 있습니다.

**[Extra 문제1]**

1. **문제정의**

이번 문제에서는, 지난 과제3에서 만든 MNIST 데이터셋을 이용하여 0~9까지의 클래스에 대한 모델을 학습하고, 성능을 평가합니다. 학습 데이터셋은 target number를 제외한 모든 클래스에서 1,000개의 샘플을 사용하여 모델을 평가하는 것 이다.

1. **학습전략**

**A. Hypothesis**

먼저, 각 클래스에 대한 Confusion Matrix와 Precision 값을 계산하였습니다. 이를 통해, 각 클래스에 대한 분류 성능을 평가할 수 있습니다. 이후, Micro & Macro averaged Precision 값을 계산하였습니다. Micro & Macro averaged Precision은 전체 클래스에 대한 평균 값을 의미합니다. 이 값을 통해, 전체 모델의 분류 성능을 평가할 수 있습니다.

**B. EDA**

본 데이터셋의 특징은 60,000장의 손글씨 이미지 데이터로 28\*28의 해상도이며 총 784개의 feature를 갖고있습니다.

1. **자신이 구현한 함수 및 클래스에 대한 설명 또는 유도 과정 작성**

1. MNIST 데이터셋을 불러오고, target number를 제외한 모든 클래스에서 1,000개의 샘플을 불러옵니다.

2. 다중 분류 모델을 선택하여, 모델을 학습합니다. 이때, 분류기는 각 클래스에 대한 분류 문제를 해결하도록 학습합니다.

3. 학습 데이터셋에서 90%의 정확도를 달성하도록 분류기를 학습합니다.

4. 테스트 데이터셋 전체를 사용하여, 각 클래스에 대한 Confusion Matrix와 Precision 값을 계산합니다. Confusion Matrix를 계산하면, 모델이 각 클래스를 얼마나 잘 분류하고 있는지를 확인할 수 있습니다. Precision은 모델이 positive로 예측한 것 중에서 실제 positive인 것의 비율을 의미합니다.

5. Micro & Macro averaged Precision 값을 계산합니다. Micro-averaged precision은 모든 클래스에 대해 TP, FP, TN, FN 값을 계산하여 평균을 구한 값입니다. Macro-averaged precision은 각 클래스의 precision 값을 평균한 값입니다.

6. 모델의 성능과 결과를 분석하여, 보고서를 작성하고 제출합니다. 이 보고서는 모델의 성능, Confusion Matrix, Precision 값, Micro & Macro averaged Precision 값 등을 포함합니다.

**4. 결과 비교 및 분석**

Chart, bar chart

Description automatically generated

**GT value별 precision은 위와 같으며 micro-average precision 결과와 macro-averaged precision결과는 아래와 같다.**

|  |  |
| --- | --- |
| Micro-averaged Precision | 0.52 |
| Macro-averaged Precision | 0.54 |

위 결과를 보면, 모델의 전체적인 분류 성능은 중간 수준입니다. Micro-averaged Precision 값이 0.52로 나타나는 것을 보면, 전체적으로는 FP(False Positive)값이 많이 발생하였다는 것을 의미합니다. Macro-averaged Precision 값이 0.54로 나타나는 것을 보면, 전체적으로는 각 클래스의 분류 성능이 비슷한 것으로 나타납니다. 이를 토대로, 모델의 성능을 개선하기 위해 다양한 실험을 진행하여 최적의 모델을 찾아낼 수 있습니다.

**5. 고찰 및 느낀점**

- 모델 평가는 모델의 성능을 정확하게 측정하여 개선하는 것이 매우 중요합니다.

- 모델 평가를 위해 여러 지표가 사용됩니다. 예를 들어, accuracy, precision, recall, F1 score 등이 있습니다.

- Confusion matrix는 모델의 성능을 평가하는 데 매우 유용합니다. 이를 통해 TP, FP, TN, FN 값을 계산하여 모델의 분류 성능을 평가할 수 있습니다.

- Micro & Macro averaged precision은 전체적인 모델 성능을 측정하기 위해 사용됩니다. 이를 통해, 전체적인 분류 성능을 측정하고 개선할 수 있습니다.

- 모델 평가는 데이터셋의 크기, 클래스의 수, 데이터 분포 등 다양한 요소에 의해 영향을 받을 수 있습니다. 이를 고려하여 모델 평가를 수행해야 합니다.

- 모델 평가를 수행하는 과정에서, 다양한 실험을 통해 최적의 모델을 찾아내는 것이 중요합니다. 이를 위해 하이퍼파라미터 튜닝, 데이터 전처리, 모델 아키텍처 변경 등 다양한 방법을 사용할 수 있습니다.