6주차 과제

Liam(황하람)

목차

1. 사용된 데이터셋
2. 주제
3. 데이터셋 분할 비율에 따른 성능 분석

- 시도한 분할 비율과 각 경우의 모델 성능 결과

- 최적의 분할 비율이라고 판단한 근거

- 분할 비율이 성능에 영향을 준 과정에 대한 해석

1. 최근접이웃(K-NN)수행 방식에 따른 성능 분석.

- 다양한 파라미터 시도와 그에 따른 성능 비교표

- 최적의 결과가 나온 설정값과 그 이유

- 성능 차이가 발생한 이유에 대한 분석

1. Perceptron, SVM, Random Forest, Naive Bayes 네 가지 알고리즘에 따른 성능 분석

- 모델별 성능 지표와 시각화 결과

- 성능 차이가 발생한 이유에 대한 해석

- 최적의 모델을 선택한 근거

1. 증강(Data Augmentation) 기법을 적용했을 때와 적용하지 않았을 때 모델 성능비교

- 증강 전/후 성능 비교표와 시각화 자료

- 어떤 증강 기법이 일반화 성능을 개선했는지 분석

- 잘못된 증강이 오히려 성능을 떨어뜨릴 수 있는 이유를 근거를 들어 설명

1. 사용된 데이터셋

* 이메일 스팸 분류

|  |  |
| --- | --- |
| **데이터셋 이름** | Spambase |
| **전체 샘플 수** | 4,601개 |
| **특징(Feature) 수** | 57개 |
| **클래스 분포** | 정상: 2,788개 (60.6%), 스팸: 1,813개 (39.4%) |

1. 데이터셋 분할 비율에 따른 성능 분석

실험 알고리즘 및 샘플코드

|  |  |
| --- | --- |
| 사용 알고리즘 | 로지스틱 회귀 |
| 샘플코드 | [https://colab.research.google.com/drive/1NE-AFhFoktwjsuYHvfaNDnteA73i3AME - scrollTo=OBkVhcgxwRtf&line=13&uniqifier=1](https://colab.research.google.com/drive/1NE-AFhFoktwjsuYHvfaNDnteA73i3AME#scrollTo=OBkVhcgxwRtf&line=13&uniqifier=1) |

- 시도한 분할 비율과 각 경우의 모델 성능 결과

|  |  |
| --- | --- |
| 분할비율 (훈련:검증:테스트) | 정확도 |
| 6:2:2 | 0.918 |
| 7:2:1 | 0.928 |
| 8:1:1 | 0.917 |

- 최적의 분할 비율이라고 판단한 근거

훈련:검증:테스트 = 7:2:1 비율에서 가장 높은 정확도(0.928)가 나타났음.

이는 모델이 충분한 학습 데이터를 확보하면서도, 적절한 양의 검증 데이터로 과적합을 방지했기 때문으로 해석됨.

- 분할 비율이 성능에 영향을 준 과정에 대한 해석

훈련 데이터가 너무 적으면 모델이 충분히 학습하지 못해 정확도가 낮아지고,

검증 데이터가 너무 적으면 모델의 성능을 제대로 평가하지 못해 잘못된 파라미터로 학습될 가능성 존재,.

따라서 7:2:1 비율은 이 두 가지 문제를 모두 피하면서 안정적인 성능 향상을 이끌어낸 비율임.

1. 최근접이웃(K-NN)수행 방식에 따른 성능 분석.

샘플코드

[https://colab.research.google.com/drive/1NE-AFhFoktwjsuYHvfaNDnteA73i3AME - scrollTo=sJNtX0VQwRya&line=14&uniqifier=1](https://colab.research.google.com/drive/1NE-AFhFoktwjsuYHvfaNDnteA73i3AME#scrollTo=sJNtX0VQwRya&line=14&uniqifier=1)

- 다양한 파라미터 시도와 그에 따른 성능 비교표

|  |  |  |  |
| --- | --- | --- | --- |
| 정규화(Y/N) | metric | K개수 | 정확도 |
| N | manhattan | 15 | 0.803 |
| Y | manhattan | 15 | 0.890 |
| Y | euclidean | 15 | 0.896 |
| Y | euclidean | 5 | 0.903 |

- 최적의 결과가 나온 설정값과 그 이유

1. 정규화 적용(Y) : 각 특징(feature) 값의 범위가 달라도 동일한 스케일로 맞춰주어, 거리 계산 시 특정 특성이 과도하게 영향 미치는 것을 방지

1. 거리선택 (Euclidean > Manhattan) : 유클리드 거리는 점 간 직선 거리를 기준으로 하므로, MNIST와 같이 연속적인 값 분포를 가진 데이터에서 더 적합.
2. K값 (5 < 15) : K가 너무 크면 다수결 투표에서 근접하지 않은 이웃까지 영향을 주어 오히려 정확도가 떨어질 수 있음.

- 성능 차이가 발생한 이유에 대한 분석

1. 정규화 N → Y : 0.803 → 0.890, 큰 차이 발생 → KNN은 거리 기반 알고리즘이므로 스케일 차이가 그대로 성능에 영향을 줌

2. 거리 metric: Manhattan → Euclidean : 0.890 → 0.896으로 소폭 개선 → MNIST 특징 벡터가 연속적이고 유클리드 거리와 잘 맞음

3. K: 15 → 5 0.896 → 0.903 : 너무 많은 이웃 포함 시 다수결 영향으로 잡음이 반영 될 수 있음, 좀 더 작은 K를 사용함으로써 잡음 영향 최소화

1. Perceptron, SVM, Random Forest, Naive Bayes 네 가지 알고리즘에 따른 성능 분석

샘플코드

[https://colab.research.google.com/drive/1NE-AFhFoktwjsuYHvfaNDnteA73i3AME - scrollTo=\_ZVLVLJXwR4K&line=27&uniqifier=1](https://colab.research.google.com/drive/1NE-AFhFoktwjsuYHvfaNDnteA73i3AME#scrollTo=_ZVLVLJXwR4K&line=27&uniqifier=1)

- 모델별 성능 지표와 시각화 결과

|  |  |  |  |
| --- | --- | --- | --- |
| 모델명 | 정확도 | 정밀도 | 재현율 |
| Random Forest | 0.951 | 0.954 | 0.927 |
| SVM | 0.923 | 0.931 | 0.882 |
| Naive Bayes | 0.827 | 0.718 | 0.963 |
| Perceptron | 0.900 | 0.857 | 0.911 |

다채로움, 그래프, 라인, 텍스트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

- 성능 차이가 발생한 이유에 대한 해석

Random Forest (최고 성능)

여러트리의 통한 평균을 사용하는 알고리즘을 통해 과적합을 방지 하였음, 따라서 정확도, 정밀도, 재현율 모두 균형 있게 높은 결과가 나왔음

Naive Bayes(최악성능)

각 특징이 독립적이라는 가정 기반으로 학습, 특징 간 의존성이 있는 경우 정확도가 낮아질수 있음, 따라서 spam문자 같은 경우 특징간 관계가 존재하기 때문에 정확도가 낮게 나옴

- 최적의 모델을 선택한 근거

RandomForest : 정확도, 정밀도, 재현율 모든 지표에서 가장 균형 있게 높은 성능을 보였음

1. 증강(Data Augmentation) 기법을 적용했을 때와 적용하지 않았을 때 모델 성능비교

샘플코드

[https://colab.research.google.com/drive/1NE-AFhFoktwjsuYHvfaNDnteA73i3AME - scrollTo=Q9x3myPnwSNT&line=5&uniqifier=1](https://colab.research.google.com/drive/1NE-AFhFoktwjsuYHvfaNDnteA73i3AME#scrollTo=Q9x3myPnwSNT&line=5&uniqifier=1)

사용된 데이터셋 : mnist

|  |  |
| --- | --- |
| **데이터셋 이름** | mnist |
| **전체 샘플 수** | 5000개 |
| **특징(Feature) 수** | 28x28 이미지 |
| **클래스 분포** | 0~9 |

- 증강 전/후 성능 비교표와 시각화 자료

|  |  |  |  |
| --- | --- | --- | --- |
|  | 이미지 확장을 통한 데이터 증강기법 사용 전/후 | | |
| 알고리즘 | 전 | 후 | 증가폭 |
| KNN | 0.941 | 0.945 | 0.04 |
| RandomForest | 0.940 | 0.946 | 0.06 |
| SVM | 0.956 | 0.957 | 0.01 |
| Perceptron | 0.870 | 0.894 | 0.024 |

스크린샷, 소프트웨어, 그래프, 라인이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

- 어떤 증강 기법이 일반화 성능을 개선했는지 분석

이미지 확대를 통해 데이터 다양성이 증가하였고, 다양성 증가로 인해 일반화 성능을 효과적으로 개선했다고 분석됨

- 잘못된 증강이 오히려 성능을 떨어뜨릴 수 있는 이유를 근거를 들어 설명

1. 원본 데이터의 패턴을 왜곡 : 모델이 잘못된 패턴을 학습하게 되어 정확도 감소

2. 노이즈 과도 추가 : 이미지 특징이 희석 분류가 어려워 질 수 있음

3. 차원 증가로 인한 학습 어려움 : 계산량이 증가하기 때문에 모델 학습 시간이 길어진다.