|  |  |  |
| --- | --- | --- |
| **| 학습주제** | | * Python에서 K-최근접 이웃(KNN) 알고리즘을 사용한 교통 혼잡 여부 예측. |
| **| 학습목표** | | * Python에서 K-최근접 이웃(KNN) 분류 모델을 학습하는 방법을 이해한다. * 시간대별 교통량 데이터를 기반으로 혼잡 여부를 예측하는 과정을 학습한다. * train\_test\_split과 StandardScaler를 사용하여 데이터를 훈련 세트와 테스트 세트로 분리하고 스케일링하는 방법을 익힌다. * 학습된 KNN 모델을 사용하여 하나의 테스트 샘플에 대한 예측을 수행하고, 실제 값과 비교하는 과정을 배운다. |
| **| 학습 개념** | K-최근접 이웃(K-Nearest Neighbors, KNN) 알고리즘은 새로운 데이터를 분류할 때 가장 가까운 K개의 이웃을 기준으로 다수를 차지하는 클래스를 예측하는 비지도 학습 방법입니다. 이 알고리즘은 훈련 데이터를 바탕으로 거리 기반 계산을 수행하여 새로운 샘플의 클래스를 예측합니다. 이 예제에서는 시간대별 교통량 데이터를 독립 변수로, 혼잡 여부를 종속 변수로 사용하여 모델을 학습하고, 선택된 샘플에 대해 혼잡 여부를 예측합니다. | |
| **| 학습 방향** | 이번 예제에서는 KNN 알고리즘은 거리 기반으로 데이터를 분류하는 방법을 다룹니다. 이 코드를 통해 시간대별 교통량 데이터를 사용하여 혼잡 여부를 예측하는 모델을 학습합니다. 데이터를 훈련 및 테스트 세트로 나누고, StandardScaler로 데이터를 표준화한 후 KNN 분류 모델을 적용하여 예측합니다. 또한, 선택된 샘플에 대한 모델의 예측값을 실제값과 비교하여 정확성을 확인할 수 있습니다. | |
| **| 요구 사항** | 1. 데이터 준비 및 전처리  - 데이터를 불러오고, 시간대별 교통량 데이터를 독립 변수로, 혼잡 여부를 종속 변수로 설정하세요.  2. 데이터 분리  - train\_test\_split을 사용하여 데이터를 훈련 데이터(70%)와 테스트 데이터(30%)로 나누세요. stratify=y로 종속 변수의 비율을 유지하세요.  3. 데이터 스케일링  - StandardScaler로 데이터를 표준화하세요.  4. 모델 생성 및 예측  - KNeighborsClassifier 객체를 사용하여 KNN 모델을 학습시키고, 테스트 셋에서 선택된 샘플의 혼잡 여부를 예측한 후, 실제 값과 비교하여 성능을 확인하세요.  . | |