|  |  |
| --- | --- |
| **| 문제의도** | * K-최근접 이웃(KNN) 알고리즘을 사용해 시간대별 교통량을 기반으로 교통 혼잡 여부를 예측하는 방법을 학습합니다. * 데이터 전처리, 스케일링, 모델 학습 및 예측 과정을 학습하고, 예측 결과를 평가하는 능력을 기르는 것이 목적입니다. |
| **| 사전지식** | * pandas, numpy, scikit-learn 라이브러리 사용법 * K-최근접 이웃(KNN) 알고리즘의 개념: 새로운 데이터를 분류할 때 가장 가까운 K개의 이웃을 기준으로 다수를 차지하는 클래스를 예측하는 모델 * 데이터 분리 및 학습: train\_test\_split을 사용해 데이터를 훈련 세트와 테스트 세트로 나누는 방법 |
| **| 문제핵심** | * 데이터 전처리: 데이터를 불러오고, \*\*독립 변수(시간대별 교통량)\*\*와 \*\*종속 변수(혼잡 여부)\*\*를 설정하는 과정 * 데이터 분리: train\_test\_split() 함수를 사용해 데이터를 훈련 세트와 테스트 세트로 분리하는 방법 * KNN 학습: KNeighborsClassifier 객체를 생성하고 모델을 학습시키는 과정 * 테스트 샘플 예측: 테스트 셋에서 하나의 샘플을 선택하고, 해당 샘플의 혼잡 여부를 예측한 후, 실제 값과 비교하는 과정 |

|  |  |
| --- | --- |
| **| 문제해설** | |
| **| 해설** | 1. 데이터 준비 및 전처리  - pandas.read\_excel()을 사용하여 엑셀 파일로부터 교통량 데이터를 불러옵니다.  - 시간대별 교통량 데이터를 \*\*독립 변수(X)\*\*로 설정하고, 혼잡 여부 데이터를 \*\*종속 변수(y)\*\*로 설정합니다.  2. 데이터 분리  - train\_test\_split()을 사용하여 데이터를 \*\*훈련 세트(70%)\*\*와 \*\*테스트 세트(30%)\*\*로 분리합니다.  - stratify=y 옵션을 사용하여 종속 변수의 비율을 유지하면서 데이터를 나누고, random\_state=42로 시드를 고정하여 일관된 결과를 재현할 수 있도록 합니다.  3. 모델 생성 및 학습  - KNeighborsClassifier 객체를 생성하고, 훈련 데이터를 통해 모델을 학습시킵니다. K개의 가장 가까운 이웃 데이터를 기반으로 새로운 데이터를 예측하는 알고리즘입니다.  4. 테스트 데이터 예측 및 성능 확인  - 테스트 세트에서 하나의 샘플을 선택하고, 해당 샘플의 혼잡 여부를 예측합니다.  - 예측된 혼잡 여부와 실제 혼잡 여부를 비교하여 모델의 예측 성능을 확인합니다. |