|  |  |
| --- | --- |
| **| 문제의도** | * Python에서 선형 회귀 모델을 사용하여 시계열 데이터를 예측하는 방법을 학습하는 것이 목표입니다. * 데이터를 훈련 세트와 테스트 세트로 나누고, 모델을 학습한 후 예측을 수행하는 과정을 통해 시계열 데이터 처리 능력을 키우는 것이 목적입니다. |
| **| 사전지식** | * Python 기초: pandas, numpy, scikit-learn 라이브러리 사용법 * 선형 회귀 모델의 기본 개념과 회귀 분석의 수학적 원리 * 데이터 전처리 및 분리: 데이터를 훈련 세트와 테스트 세트로 나누는 방법 |
| **| 문제핵심** | * numpy.arange()와 reshape()을 사용하여 날짜 인덱스를 독립 변수로 설정하는 과정 * train\_test\_split()을 사용하여 데이터를 훈련 세트와 테스트 세트로 나누는 방법 * LinearRegression 객체를 사용하여 선형 회귀 모델을 학습시키는 과정   predict() 함수를 사용하여 모델이 새로운 데이터를 예측하는 과정 |

|  |  |
| --- | --- |
| **| 문제해설** | |
| **| 해설** | 1. 데이터 준비 및 전처리  - pandas를 사용하여 엑셀 데이터를 불러오고, 날짜를 독립 변수로 설정한 후, 8시 교통량 데이터를 종속 변수로 설정합니다.  2. 데이터 분리  - train\_test\_split을 사용하여 데이터를 훈련 세트와 테스트 세트로 분리합니다. 이 과정에서 훈련 데이터는 모델 학습에 사용되고, 테스트 데이터는 모델 성능 평가에 사용됩니다.  3. 모델 학습  - LinearRegression 객체를 생성하고, fit() 메서드를 사용해 훈련 데이터를 기반으로 선형 회귀 모델을 학습시킵니다. 이를 통해 모델이 데이터의 패턴을 학습합니다.  4. 마지막 날의 교통량 예측  - 학습된 모델을 사용해 마지막 날의 교통량을 예측합니다. predict() 함수를 사용하여 모델이 예측값을 생성합니다.  5. 결과 비교  - 마지막 날의 실제 교통량과 모델의 예측값을 비교하여, 예측 성능을 평가하고 모델의 정확도를 확인합니다. |