1주차 예비 보고서

전공: 컴퓨터공학 학년: 2학년 학번: 20221559 이름: 박준우

1-1. Supervised Learning(지도 학습)과 Unsupervised Learning(비지도 학습)의 주요 차이점은 학습 데이터의 구성 요소와 이를 통해 해결할 수 있는 문제의 종류이다. 지도 학습은 학습 데이터에 입력 데이터(features)뿐만 아니라, 입력에 대한 정답(label) 이 함께 제공된다. 이에 반해 비지도 학습은 정답이 없이, 구분되지 않은 입력 데이터만 주어진다.

지도 학습의 목표는 주어진 데이터에서 정답을 예측하는 능력을 학습하는 것이다. 예를 들어 학습 데이터의 정답이 카테고리의 구분으로 주어졌다면 예측한 정답도 카테고리의 구분이 될 것이고, 어떠한 수치로 주어졌다면 결과도 수치가 되는 것이다.

반면 비지도 학습의 목표는 정답 없이 주어진 데이터 내부의 숨겨진 구조나 패턴을 발견하는 것이다. 대표적으로 수행할 수 있는 작업은 주어진 데이터의 특성을 기준으로 그룹을 나누거나, 데이터의 분포를 이해하는 것 등이 있다. 예를 들어, 고객 데이터를 비지도 학습 모델에 적용해 구매 행동이 유사한 고객들을 그룹화 해서, 각 그룹의 특성을 파악하고, 각 그룹을 적절히 분할하는 데 사용할 수 있다.

이처럼 모델마다 수행 가능한 문제와 결과의 종류가 다르기 때문에, 모델이 잘 학습되었는지 판단하는 기준 및 최적화 목표 또한 다르다. 지도 학습 모델들은 주어진 정답을 얼마나 정확히 예측했는지, 또는 정답에 얼마나 가까운지 등을 수치화 해 평가한다. 이런 평가를 바탕으로 지도 학습 모델은 새로운 데이터에서도 높은 정확도로 정답을 예측하는 것을 목표로 한다. 반면 비지도 학습은 데이터의 구조나 패턴을 학습한 결과를 기준으로 평가한다. 예를 들어 그룹간의 구분이 얼마나 명확한지, 각 그룹 내 데이터 간의 유사성이 높은지 등을 이용해 평가할 수 있다. 결과적으로 비지도 학습은 새로운 데이터가 들어왔을 때, 데이터의 특성을 잘 분석하고 분류하는 것을 목표로 한다.

1-2. 모델을 학습한다는 것이 무엇을 의미하는지 먼저 이해하는 것이 중요할 것 같다. 앞서 설명했듯 인공지능 모델을 학습시킨다는 것은 입력 데이터 내의 연관성, 패턴 등을 분석해 문제 해결에 사용할 수 있는 규칙이나, 대응관계를 찾아가는 과정을 의미한다.

모델 학습의 첫 단계는 학습에 사용할 데이터의 수집과 준비이다. 현실에서 수집한 데이터는 학습에 바로 사용하기에 다양한 제한사항이 있을 수 있는데, 학습을 위해 사용할 수 있도록 전처리 해 적합한 형태로 만드는 작업이 필요합니다. 누락된 값 처리, 이상치 제거, 정규화 등의 전처리 과정을 거친 후 목적에 맞게 사용할 수 있도록 분리하게 된다.

그 다음 모델의 사용목적과 주어진 데이터의 특성을 고려하여 적합한 모델을 선택한 후, 모델 학습을 실시한다. 이후는 학습을 시킨 모델이 얼마나 잘 학습되었는지 검증하기 위해 앞서 분리해 놓은 테스트 데이터를 통해 확인해 보거나, 실제 예측 작업을 수행하여 결과를 분석한다. 이 과정에서 모델의 성능이 만족스럽지 못 할 경우 학습 데이터를 추가하거나, 줄이는 등의 작업이나 모델 내부의 인자들을 잘 조절하는 등 최적화 과정을 거쳐 원하는 결과가 나오도록 수정하는 과정을 거친다. 최종적으로는 모델이 기대하는 성능을 달성할 때까지 이런 과정을 반복하는 것이 모델을 학습한다는 것에 대한 과정이라고 생각한다.

2-1. Numpy는 빠른 속도의 배열 연산과 수치의 계산을 위한 도구로, 데이터의 기초적인 처리와 계산을 담당한다. Pandas는 테이블 형태의 데이터를 효율적으로 분석하고 조작하기 위해 사용되며, 데이터프레임을 활용한 강력한 기능을 제공한다. Matplotlib는 데이터를 시각적으로 표현해주는 라이브러리로, 주로 결과의 직관적 이해를 돕는다.

Numpy는 다차원 배열인 ndarray라는 객체를 중심으로 여러 기능들을 제공한다. 우선 반복문 없이 배열 전체를 빠르게 처리할 수 있는 연산을 지원한다. 예를 들어, 배열의 모든 요소에 동일한 연산을 수행하거나, 배열 간의 연산을 효율적으로 처리할 수 있다. 또한, Numpy는 선형대수 계산(행렬, 벡터 연산)을 포함한 다양한 수학적 기능을 제공하여 데이터 분석과 머신러닝에서 자주 필요한 수학적 작업을 쉽게 수행할 수 있다. 크기가 다른 배열 간에도 연산을 자동으로 확장하여 처리 가능하며, 기본적인 통계 함수, 푸리에 변환, 난수 생성 등 다양한 기능을 포함하고 있다. Numpy는 Pandas, Matplotlib, 더 나아가 Scikit-learn 등 다른 데이터 분석 라이브러리의 기반으로, 효율적인 처리와 계산을 가능하게 하는 것이 특징이다.

Pandas는 데이터프레임과 시리즈 같은 고급 데이터 구조를 제공한다. 데이터프레임은 행과 열로 구성된 구조로, 엑셀이나 SQL 테이블처럼 데이터를 쉽게 다룰 수 있다. Pandas는 데이터를 선택, 필터링, 정렬, 병합, 그룹화 등의 작업을 간단하게 수행할 수 있으며, 이를 통해 결측값 처리와 데이터 정제에서 사용하기 편리하다. 또한, CSV, Excel, JSON, SQL 등 다양한 파일 형식의 데이터를 읽고 저장하는 입출력 기능을 지원해 데이터 분석의 유연성을 높인다. Pandas는 기본적인 통계 연산(평균, 분산, 빈도수 계산 등)을 포함하며, 날짜/시간 데이터 처리와 시계열 분석에도 적합하다. 특히, Pandas는 Matplotlib와 연동하여 데이터를 바로 시각화할 수 있는 기능을 제공하며, 데이터 분석과 머신러닝의 사전 작업에 필수적인 도구로 널리 활용된다.

Matplotlib는 다양한 유형의 그래프와 플롯을 생성할 수 있는 도구를 제공한다. 선 그래프, 막대 그래프, 히스토그램, 산점도, 파이 차트 등 다양한 시각화 형식을 지원하며, 데이터를 직관적으로 표현하는 데 유용하다. 핵심 모듈인 pyplot은 간단한 명령어를 통해 그래프를 빠르게 생성할 수 있는 MATLAB 스타일의 인터페이스를 제공한다. 또한, 그래프의 크기, 색상, 폰트, 축 레이블, 제목 등을 세부적으로 커스터마이징이 가능하다. Matplotlib는 Pandas와 연동하여 데이터프레임을 바로 시각화할 수 있으며, 데이터를 분석한 결과를 그래프로 표현하는 데 필수적인 도구로 활용된다.

3. scikit-learn

1) Linear Regression (선형 회귀)

Linear Regression은 독립변수와 예측값 (종속변수) 간의 관계를 분석하여 가중치를 학습하고, 이를 기반으로 이 데이터들의 특징을 나타내는 최적의 직선(또는 초평면)을 찾아 새로운 데이터에 대한 예측을 수행하는 모델이다.

변수명 = LinearRegression() 으로 선언하고, 전달해 줄 수 있는 주요 파라미터는

1. fit\_intercept : 독립변수에 곱해진 가중치가 아닌 상수를 0으로 고정할 지의 여부를 설정
2. copy\_X : 입력 데이터를 복사해 사용할 지의 여부를 설정

이 있다.

Model.fit(X, y) / y = Model.predict(X) 으로 학습, 예측을 실시할 수 있다.

또, 학습 시 각 샘플에 대한 가중치를 제공해 주어서, 데이터 간의 중요도를 설정할 수 있다.

사용 시 fit(X, y, sample\_weight = [])로 전달해 주면 된다.

학습이 완료된 후 .coef\_, intercept\_로 최종 가중치를 확인할 수도 있다.

2) Decision Tree (의사 결정 트리) – classification / regression

데이터를 조건에 따라 반복적으로 분리하여 트리 구조를 생성해서 분류, 연속적 값을 예측 가능한 모델이다. 각각 DecisionTreeClassifier, DecisionTreeRegressor 클래스로 정의 되어 있다.

비선형적인 데이터에 대응하기 좋으나, 학습 데이터에 과적합될 가능성이 높은 모델이다.

과적합을 방지하는 데 사용할 수 있도록 선언 시 파라미터로 트리의 최대 깊이, 분할 기준, 독립변수 반영 최대개수 제한 등을 설정할 수 있다.

LinearRegression 모델과 fit, predict는 기초적인 용법은 동일하다.

3) Random Forest – classification / regression

의사 결정 트리의 과적합 문제를 해결하기 위해서, 무작위로 사용할 학습 데이터와 독립변수를 뽑아 여러 의사 결정 트리를 만들어 트리 간의 다양성을 확보한다. 이후 분류 모델에서는 각 트리들의 결과 중 다수결로, 회귀 모델에서는 결과의 평균치로 최종 예측 결과를 낸다.

각 모델은 RandomForestClassifier, RandomForestRegressor 클래스로 정의 되어 있다.

의사 결정 트리를 선언할 때 넣던 파라미터에 더해, 생성할 트리의 개수도 설정할 수 있다.

기초적인 fit, predict 용법은 마찬가지로 동일하다.