중간고사 일정: 10/23 (목), 10:30 ~ 12:30 (2hr)

- 범위: 강의노트 Lecture 1 ~ Lecture 6 (단, Softmax-CrossEntropy 제외)

- 객관식 20문항
  - 각 문항 당 4점
  - 오답 시 -4점

- 주관식 2문항 (서술형)
  - 각 문항 당 10점
  - 서술 시에 필수 키워드 중심으로 배점

Lab 1

- CSV 파일 불러오기

  Python 표준 라이브러리로 불러오기
  - NumPy로 불러오기
  - Pandas로 불러오기

- Pima Indians 데이터셋

  이 데이터셋은 피마 인디언(Pima Indian) 여성들의 의학적 기록을 담고 있으며,
  환자가 5년 이내에 당뇨병이 발병할지를 예측하는 이진 분류 문제

Lab 1

- 기술통계 (Descriptive Statistics)
  목적: 데이터를 요약·정리하여 전체적인 경향이나 특징을 파악
  예시: 평균, 중앙값, 표준편차, 그래프(히스토그램, 상자그림 등)

- 데이터를 직접 확인하는 것은 매우 중요합니다. Pandas를 활용하면 다음과 같은 탐색을 할 수 있
- 데이터 미리보기 (head, tail)
- 데이터 차원 확인 (shape)
- 속성별 데이터 타입 확인 (dtypes)
- 통계량 (describe: 평균, 표준편차, 최소/최대, 사분위수)
- 클래스 분포 (class balance)
- 속성 간 상관관계 (correlation matrix)
- 왜도(Skewness) 확인

Lab 2

● 시각화 (Visualisation)

데이터를 잘 이해해야 머신러닝 알고리즘에서 좋은 결과를 얻을 수 있습니다.
가장 빠른 방법은 데이터 시각화입니다 (Pandas를 활용해 데이터를 시각화)

단변량(Univariate) 플롯
데이터 속성 하나씩을 독립적으로 살펴보는 방법 3가지:

- 히스토그램 (Histogram): 분포 형태를 추정 및 이상치 확인 가능
- 밀도 플롯 (Density Plot): 매끄러운 분포 확인가능
- 상자 (수염) 그림(Box Plot): 중앙값, 사분위수, 데이터분포 범위, 이상치 직관적으로 확인

다변량(Multivariate) 플롯

- 상관행렬 (Correlation Matrix Plot): 변수간 상관관계 확인, 대각선은 항상 1
- 산점도행렬 (Scatter Plot Matrix): 변수 쌍마다 관계 확인

Lab 2

● 데이터 전처리 (Data Preparation)

머신러닝에서는 데이터 전처리가 거의 항상 필요합니다.

알고리즘마다 데이터에 대한 가정이 다르므로 다양한 변환을 시도하는 것이 좋습니다.

1) MinMaxSaler (Rescale: 0~1 범위)
2) Standardisation: 평균=0, 표준편차=1
3) Normalisation: 벡터 길이를 1로 변환

4) Binarisation: 임계값 기준 0/1 변환

Lab 3

● Feature Selection

머신러닝 모델 성능은 어떤 특징(feature)을 사용하는지에 크게 좌우됩니다.
관련 없는 특징이 포함되면 정확도가 떨어지거나 과적합이 발생할 수 있습니다.

- 과적합 감소 → 불필요한 데이터 감소
- 정확도 향상 → 혼란스러운 데이터 제거
- 학습 시간 단축 → 데이터 차원이 줄어듦

Lab 3

● Feature Selection

머신러닝 모델 성능은 어떤 특징(feature)을 사용하는지에 크게 좌우됩니다.
관련 없는 특징이 포함되면 정확도가 떨어지거나 과적합이 발생할 수 있습니다.

- 과적합 감소 → 불필요한 데이터 감소
- 정확도 향상 → 혼란스러운 데이터 제거
- 학습 시간 단축 → 데이터 차원이 줄어듦

예) 카이제곱 검증, SelectKBest, RFE, PCA, Feature Importance

예) 산업현장에서 현재 Boruta 활용 중

Lab 3

● Resampling

훈련 데이터만으로 성능을 평가하면 과적합 위험이 있습니다.

따라서 리샘플링 기법을 사용하여 모델이 새로운 데이터에 대해 얼마나 잘 일반화되는지 추정합니다

- Train/Test 분할 → 데이터 67% train, 33% test: 분할에 따른 성능편차가 큼
- K-fold 교차검증 → 데이터를 k개의 fold 로 나누어 k번 학습/검증, 일반적으로 가장 많이 활용
  실무나 논문에서 k=10 혹은 k=5 많이 쓰임
- LOOCV(Leave-One-Out Cross Validation) → 많은 연산량으로 실무에서는 K-fold 를 주로 활

➔ 가장 일반적이고 무난한 방법으로서, 10-fold cross validation 을 활용

Lab 4

● 평가지표 (Evaluation Metrics)

머신러닝 알고리즘의 성능 평가 지표는 중요합니다.

- 어떤 지표를 선택하느냐에 따라 알고리즘의 성능 비교가 달라짐
- 최종적으로 어떤 모델을 선택할지 결정에 큰 영향

→ 예측문제는 회귀(Regression)와 분류(Classification)로 문제를 구분

→ Pima Indians Diabetes Dataset 을 활용한 분류문제에 대한 다양한 평가 지표

예) Accuracy, Logloss, AUROC, Confusion Matrix,
Classification Report(Precision, Recall, F1)

Lab 4

- 알고리즘 성능 비교 (Comparing ML Algorithms)
  다양한 알고리즘을 시범 적용하여 성능 비교

- Logistic Regression, Linera Discriminant Analysis, …
- K-Nearest Neighbour, Support Vector Machine,
- Decision Tree(CART), Naïve Bayes, …

  이외) Random Forest, Gradient Boosting, XGBoost …, Multi-Layer Perceptron (ML

→ 박스플롯(Boxplot) 등 시각화를 통해 모델별 정확도 분포 비교 가능

- 최종적으로,

다양한 평가 지표(Accuracy, LogLoss, AUC, Confusion Matrix, Report)를 활용해 모델 성능
다각도로 검증해야 함, Lab 예제에서는 Logistic Regression와 LDA가 성능우위

Lab 5

- Automating ML Algorithms
- Pipelining: 여러 전처리 단계 + 모델 학습을 하나의 체인으로 묶어줌
- FeatureUnion: 여러 특징 추출 기법을 합쳐 하나의 데이터셋으로 만들어줌

- Ensemble(앙상블)
- Bagging: 여러 샘플 → 여러 모델 학습 → 평균 (예: Random Forest)
- Boosting: 이전 모델의 오류(에러)를 보완하는 방식 (예: XGBoost)
- Voting: 서로 다른 모델들의 예측을 결합 (실무에는 거의 활용안됨)

Lab 5

- Tuning Hyperparameters

- 모델 성능 최적화를 위해 하이퍼파라미터 탐색 수행
- Grid Search: 지정된 파라미터 그리드 전수 탐색
- Random Search: 랜덤 샘플링 기반 탐색

- Saving and Loading Models (모델 저장과 불러오기)

최종적으로

- **Pipeline → 데이터 누수 방지 + 자동화**
- **FeatureUnion → 여러 특징 추출 기법 결합 가능**
- **Ensemble → Bagging, Boosting, Voting으로 성능 향상**
- **Tuning → GridSearch, RandomSearch로 최적하이퍼파라미터**
- **Saving → Pickle 등으로 모델 저장 및 재사용 가능**

Lab 6 MLP

● 모델 정의

- 입력변수: 8개
- 첫번째 층: 뉴런 12개, 활성화 함수 Relu
- 두번째 층: 뉴런 8개, 활성화 함수 Relu
- 출력층: 뉴런 1개, 활성화함수 Sigmoid

● 모델 컴파일

- 손실함수: binary_crossentropy
- 최적화 알고리즘: Adam, -metric=[‘accuracy’]

● 모델 학습(fit) 및 평가 (evaluate)

Lab 7 AutoEncoder

- 오토인코더는 입력을 다시 출력으로 복원하는 비지도 학습 기반 신경망 구조
  핵심 아이디어는 입력을 낮은 차원으로 압축(Latent Space, Coding)하고, 그 정보만으로 원래 입력을 복
- Reconstruction Loss 를 통한 학습
- 이 과정을 통해 모델은 데이터의 중요한 특징을 학습하는 것

구분	구조적 특징	학습 목표	주요 장점
Simple Autoencoder	입력=출력	데이터 압축 및 복원	의미있는 특징 추
Denoising Autoencoder	입력에 노이즈 추가	노이즈 제거 학습	불변적 특징 학습
Sparse Autoencoder	L1 Regularization 추가	희소 표현 학습	효율적 표현, 과

● Representation Learning

Lab 8 CNN

- CNN 모델 정의
  - •Conv2D (32 filters, 5×5, relu, padding=same),
    입력과 출력의 공간적 크기를 동일하게 유지하기 위해 입력의 가장자리에 0으로 패
    추가하는 합성곱 설정, 이를 통해 CNN 은 가장자리 정보까지 보존하며, 네트워크가
    공간차원이 축소되지 않음
  - •MaxPooling2D (2×2)
  - •Conv2D (64 filters, 2×2, relu, padding=same)
  - •MaxPooling2D (2×2)
  - •Dropout (0.25)
  - •Flatten (2D → 1차원 벡터 변환)
  - •Dense (1000, relu)
  - •Dropout (0.5)
  - •Dense (10, softmax)
- 모델 컴파일
  - 손실함수: categorical_crossentropy
  - 최적화 알고리즘: Adam, -metric=['accuracy']
- 모델 학습(fit) 및 평가 (evaluate)