

**SK네트웍스 Family AI 과정 15기  
 데이터 전처리 학습된 인공지능 모델**



| **산출물 단계** | 데이터 전처리 |
| --- | --- |
| **평가 산출물** | 학습된 인공지능 모델 |
| **제출 일자** |  |
| **깃허브 경로** | 깃허브 주소 기재 |
| **작성 팀원** |  |

1. **모델 목적**: (예: 사용자 질문에 대해 적절한 상담 주제를 분류하기 위한 모델)
2. **모델 아키텍처 설계**

* 선정 모델: 예) BERT 기반 분류 모델
* 아키텍처 개요:

| **계층** | **구성 요소** | **역할** |
| --- | --- | --- |
| 입력층 | Tokenizer + Embedding | 문장 벡터화 |
| 인코더 | Transformer Encoder Blocks (12층) | 의미 표현 학습 |
| 출력층 | Dense Layer + Softmax | 다중 클래스 분류 |

* 아키텍처 시각화: (구조 다이어그램 삽입 또는 링크)
* 설계 근거  
   예) Transformer 기반 모델은 문맥 이해에 강하며, 긴 문장에 대한 표현력이 뛰어남

1. **모델 학습 요약**

* 학습 데이터 수: 10,000건
* 검증 데이터 수: 2,000건
* 평가 데이터 수: 1,000건
* 성능 평가 결과:

| **지표** | **값** |
| --- | --- |
| Accuracy | 91.4% |
| Precision | 92.0% |
| Recall | 91.2% |
| F1 Score | 91.6% |

* 일반화 성능 평가:  
  + 미검증 데이터셋(Test set)에 대한 성능 평가 결과 포함
  + 과적합 방지를 위해 조기 종료, Dropout 등 사용
  + 과소적합 발생 없음 (훈련 정확도와 테스트 정확도 유사)

1. **저장 및 배포**

* 저장 형식:

| **항목** | **설명** |
| --- | --- |
| 저장 파일명 | model\_final\_v1.pt |
| 저장 형식 | PyTorch .pt 파일 (또는 TensorFlow SavedModel) |
| 저장 방법 | torch.save(model.state\_dict(), 'model\_final\_v1.pt') |
| 모델 불러오기 코드 예시 | model.load\_state\_dict(torch.load('model\_final\_v1.pt')) |

* 모델 사양 요구 사항:  
  + 프레임워크: PyTorch 2.0
  + GPU/CPU 호환 여부: GPU 사용 시 학습시간 단축, CPU에서도 추론 가능
  + 환경 설정 파일: requirements.txt 포함 (예: transformers==4.38, torch==2.1)
* 모델 테스트:  
  + 모델 적재 및 추론 테스트 완료
  + Inference 예시:  
      
     입력: “내일 날씨 알려줘”  
     출력: “날씨” (예측 정확)

1. **종합 평가 및 활용 방안**

* 모델 안정성: 저장/로드 반복 시 정확도 재현 가능 (±0.1%)
* 일반화 가능성: 미사용 데이터셋에서도 높은 성능 유지
* 재사용성: 모델 저장 및 배포 시 용량 420MB, 추론 평균 시간 0.3초/건
* 향후 활용: API 서버에 탑재, 챗봇 응답 분류, 유사 문장 검색 등에 사용 예정

1. **추가 기재**

* 저장된 모델 파일 위치 또는 URL:링크 주소기재
* 모델 저장/불러오기 코드 (선택)
* 학습 로그 또는 스크린샷 (TensorBoard 등)