

(MeloMatch)

컴퓨터비전과 자연어 처리를 활용한 개인화된 음악추천

동양 미래 대학교
자동화공학과
자동화 프로젝트1[C1]

20243696 이석현, 20243703 김해현



(MeloMatch)

Contents

01 과제 소개 및 핵심 기술 소개

- 과제 소개
- 프로젝트 목표
- CORETECHNOLOGY
- 시스템 구조도
- 주요 흐름

02 입력 및 인식

- 개발 순서
- 컴퓨터비전
- 자연어 처리
- 모델 병합
- UI 개발
- 추후계획

03 음악 알고리즘 및 AI음악작곡

- 개요
- 기술적 배경
- 알고리즘 구조
- 전체 흐름도
- 가중치 설명
- 부분 시연

04 작품 시연

과제 소개



현대 사회에서 사람들은 감정과 상황에 맞는 음악을 찾는 데 어려움을 겪고 있습니다. 다양한 음악 플랫폼이 존재하지만, 대부분의 추천 시스템은 사용자의 감정 상태나 나이와 같은 개인적 요소를 충분히 고려하지 못하고 있습니다. 이로 인해 음악 추천이 때로는 사용자의 기대와 맞지 않는 경우가 많습니다. 이러한 변화에 발맞춰, 저희는 "MeloMatch" 프로젝트를 통해 사용자의 감정 상태를 인식하고, 그에 맞는 음악을 추천하여 개인의 생활에 새로운 가치를 제공하고자 합니다.

PROJECT

프로젝트 목표

01

컴퓨터 비전과 자연어
처리의 결합
컴퓨터 비전으로 사용자의
나이와 성별을 파악하고 자
연어 처리(NLP)를 활용하여
사용자의 감정상태를
확인하는 시스템 구성

02

감정에 맞는 MIDI 음악 작곡
기능을 제공하여 사용자의
정서적 상태를 반영한 창의
적인 음악 생성.

03

유튜브 API를 활용하여 검
색 쿼리를 자동 생성하고
적합한 음악을 추천, 재생
하는 시스템 구축.

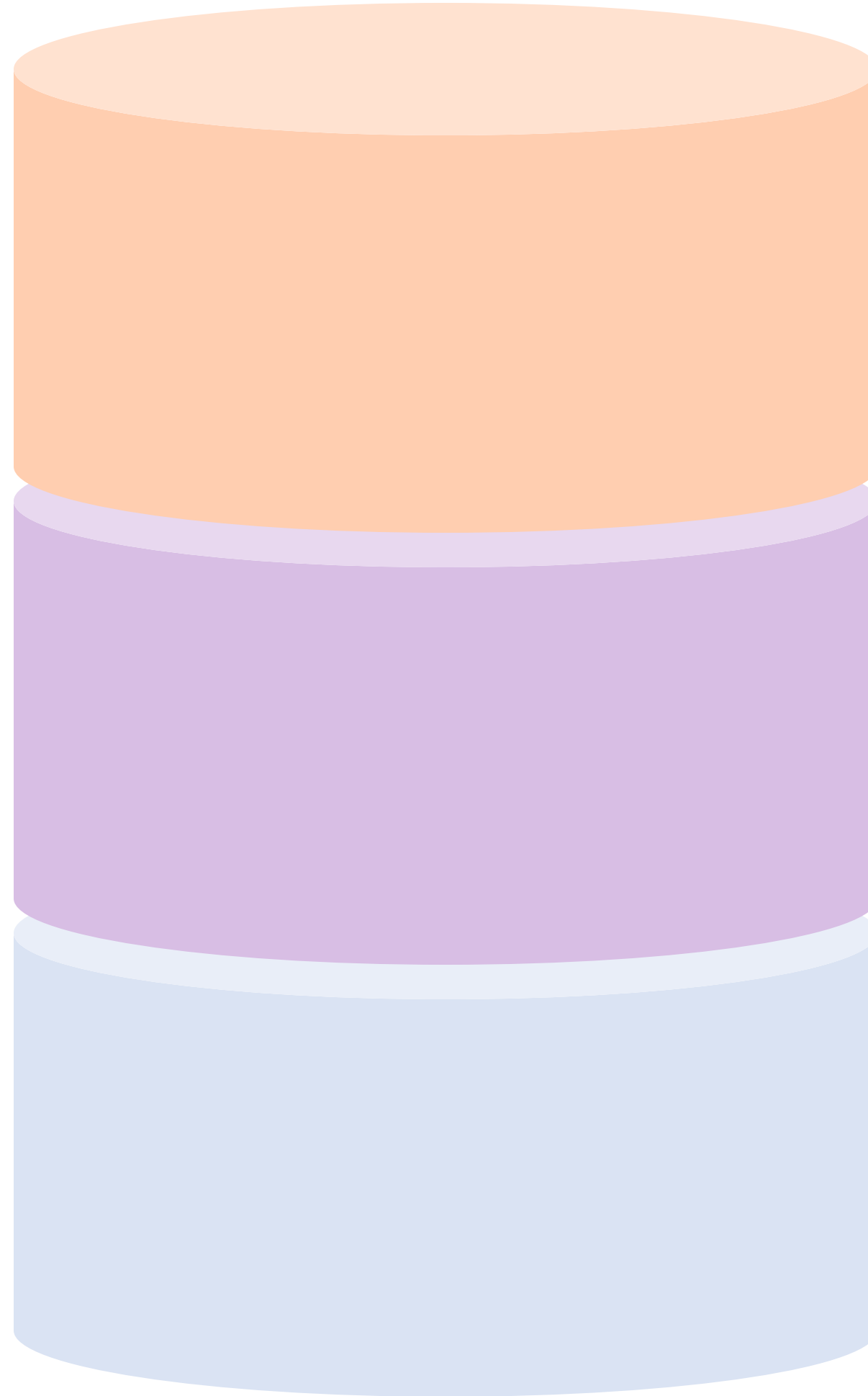
CORETECHNOLOGY

자연어 처리

사용자의 목소리를
인식해 감정을 분석
합니다.

컴퓨터 비전을 활용한 얼굴인식

사용자의 얼굴을 인식하여
성별, 나이 인식합니다.

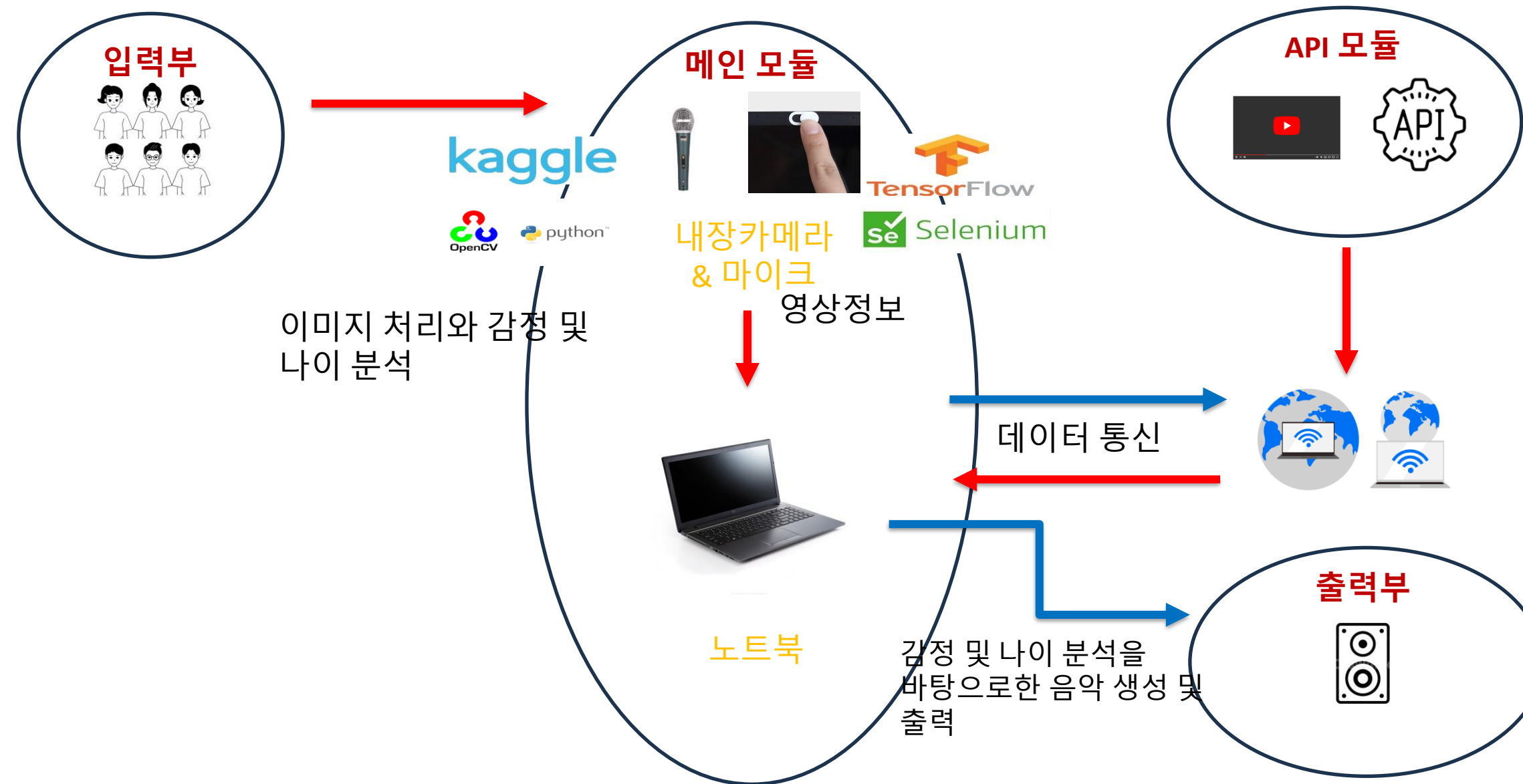


음악 추천 과 AI음악 생성

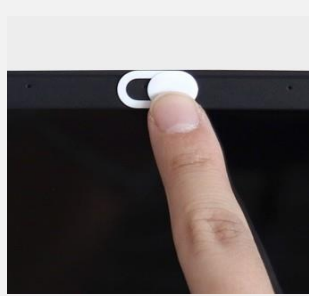
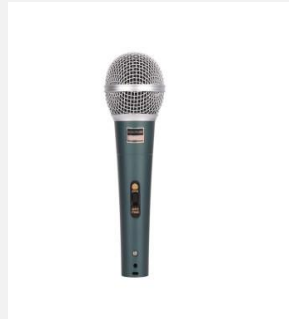
인식된 데이터에 맞는 음악
데이터 베이스에서
선별하여 추천하고 상황에
맞는 음악을 생성합니다.



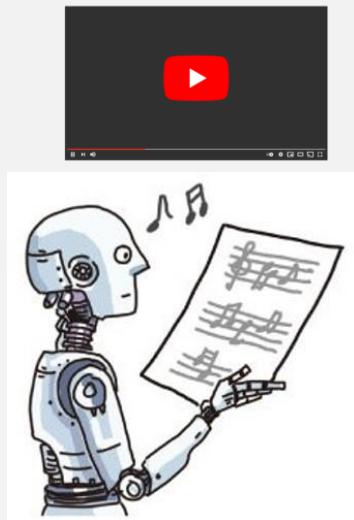
시스템 구조도



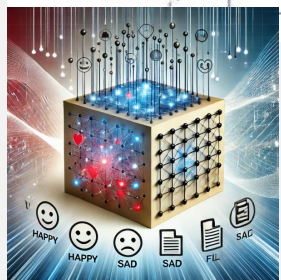
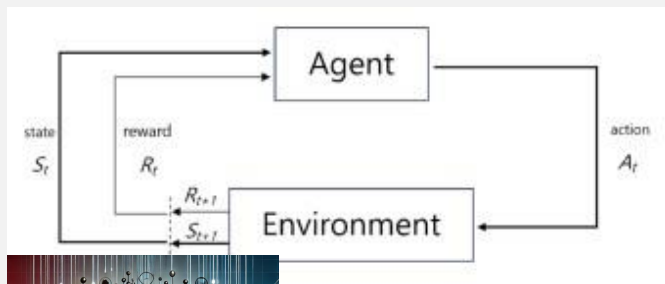
주요 흐름



- OpenCV를 활용하여 실시간 얼굴 데이터를 수집
- 사용자의 음성을 녹음하여 음성 감정을 분석
- Google Speech-to-Text API를 사용해 음성을 텍스트로 변환
- 변환된 텍스트 데이터를 KcELECTRA 모델로 분석하여 감정을 예측
- 음성 감정과 텍스트 감정과 얼굴 데이터를 통합해 최종 감정을 도출



- YouTube API를 통해 감정 및 사용자의 나이에 따라 음악을 검색
- Selenium으로 브라우저를 열어 추천된 음악을 자동 재생
- MIDIUtil 라이브러리를 사용해 감정에 맞는 음계와 악기를 조합하여 작곡
- 감정별로 다른 템포, 스케일, 악기를 활용



- JSON 파일로 가중치를 관리하며, 감정별로 초기값을 설정
- 프로그램 실행 시 기존 데이터가 로드되며, 피드백에 따라 업데이트
- 사용자가 추천 음악에 대한 만족도를 평가
- 긍정 피드백 시 해당 감정의 가중치 증가, 부정적 피드백 시 감소

PROJECT

컴퓨터 비전 인식 사용 기술



python

주요 개발 언어로 얼굴 감정 데이터를 결과와 통합하여 주요 감정 상태 도출



DeepFace

나이, 성별 분석



MTCNN

얼굴 탐지 및 좌표 추출

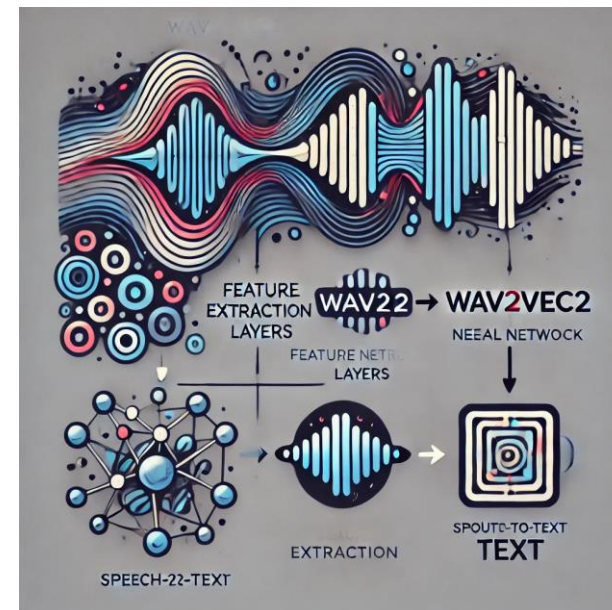
PROJECT

자연어 처리 사용 기술



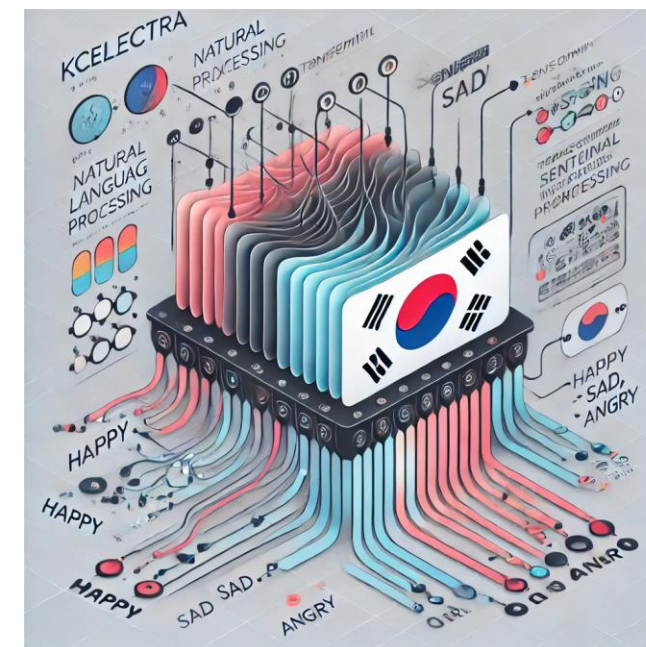
Google Speech-to-Text

음성을 텍스트로
변환



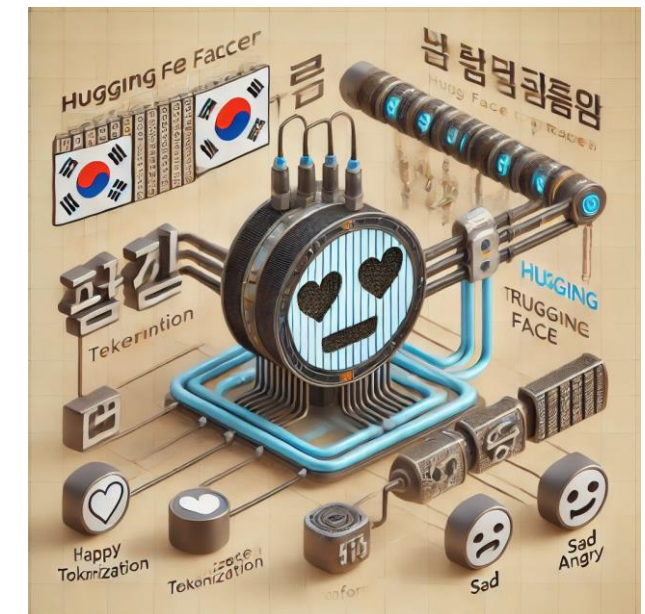
Wav2Vec2

음성 데이터를 텍
스트로 변환하며
음성의 감정 분석
에 사용



KcELECTRA

변환된 텍스트 데
이터를 감정 분석
에 활용



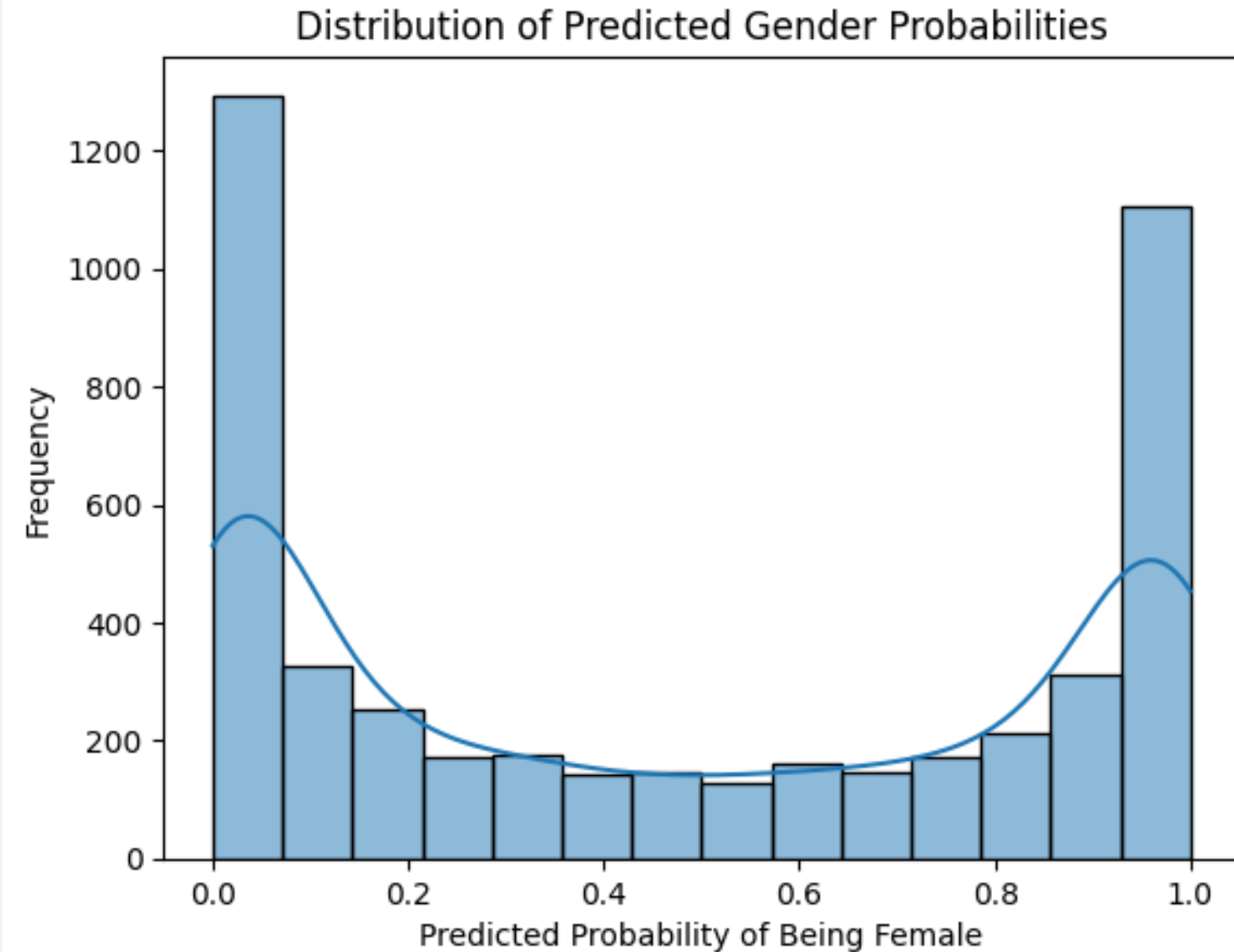
Hugging Face
Transformers

사전 학습된 NLP 모델을
활용하여 전이학습을 통해
감정분석 및 처리

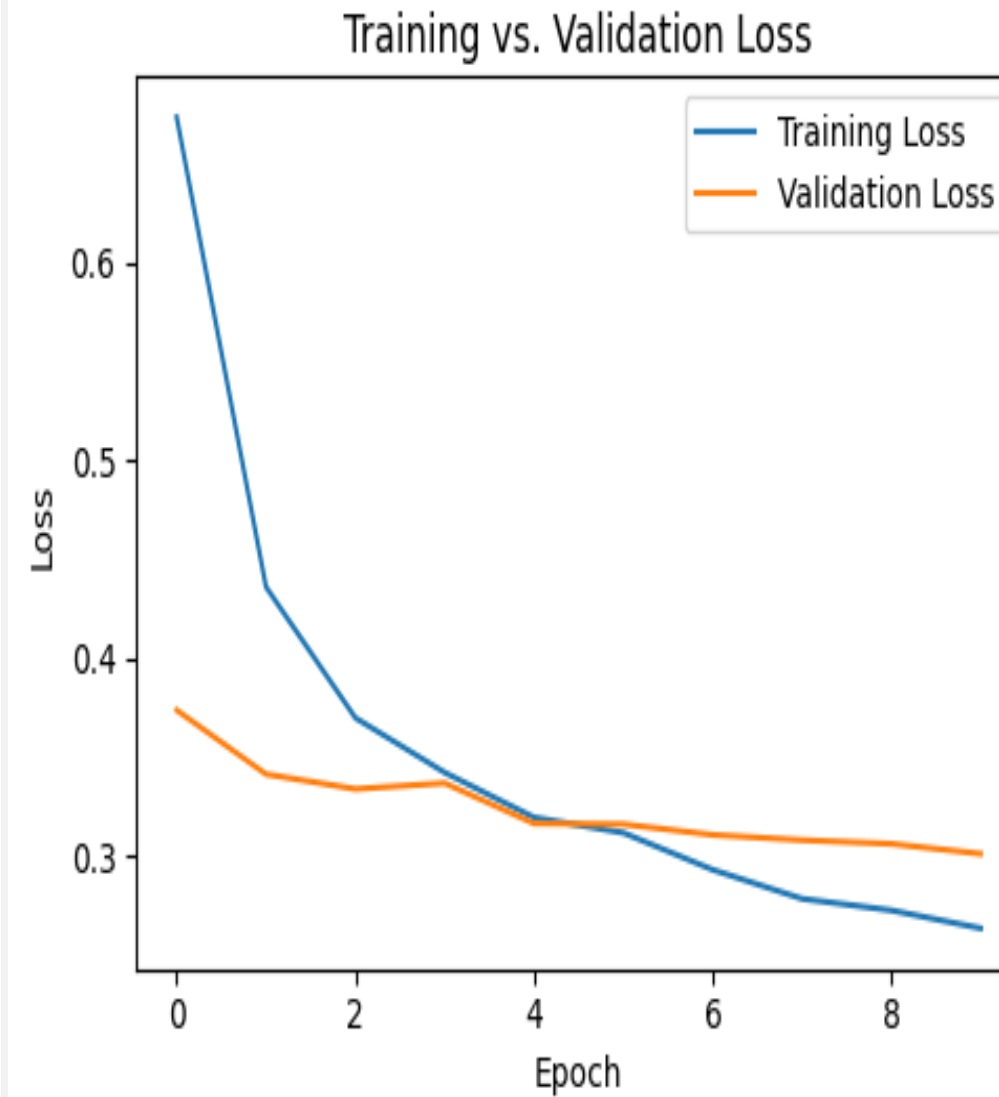
컴퓨터 비전인식 및 자연어 처리



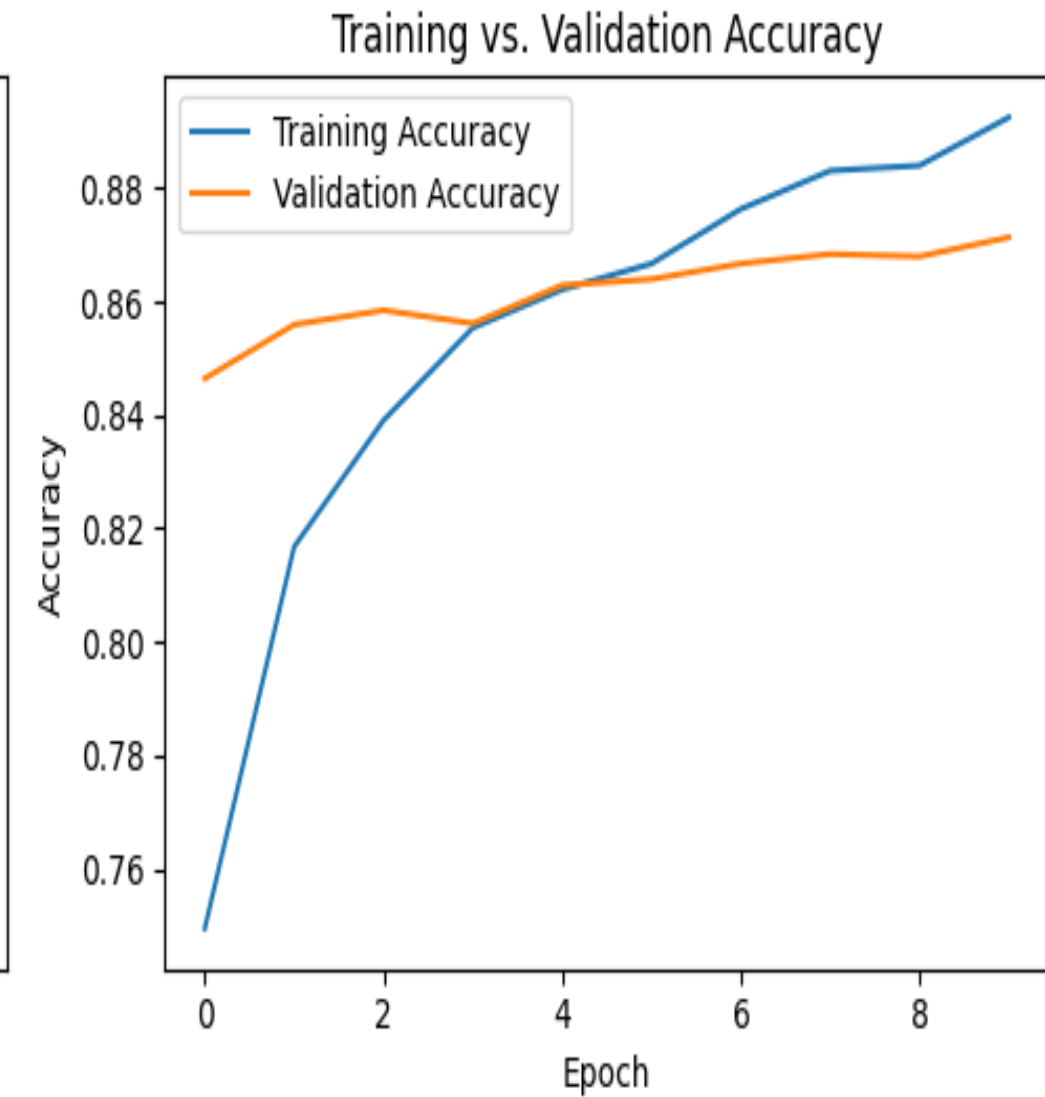
성별 학습



예측된 성별의 확률 분포
x축-0은 남자 1은 여성
y축-각 확률 구간의 예측 값들의 빈도

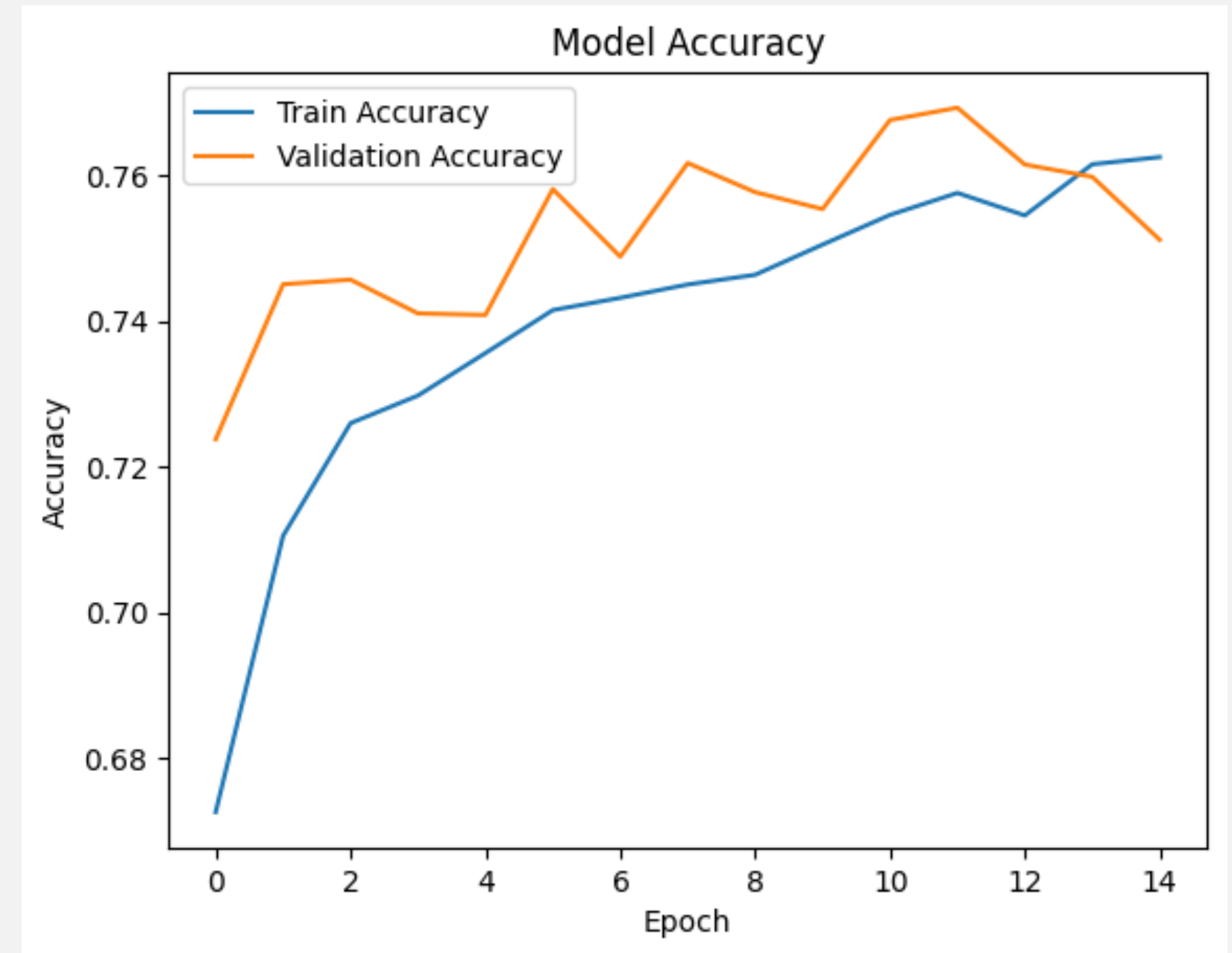
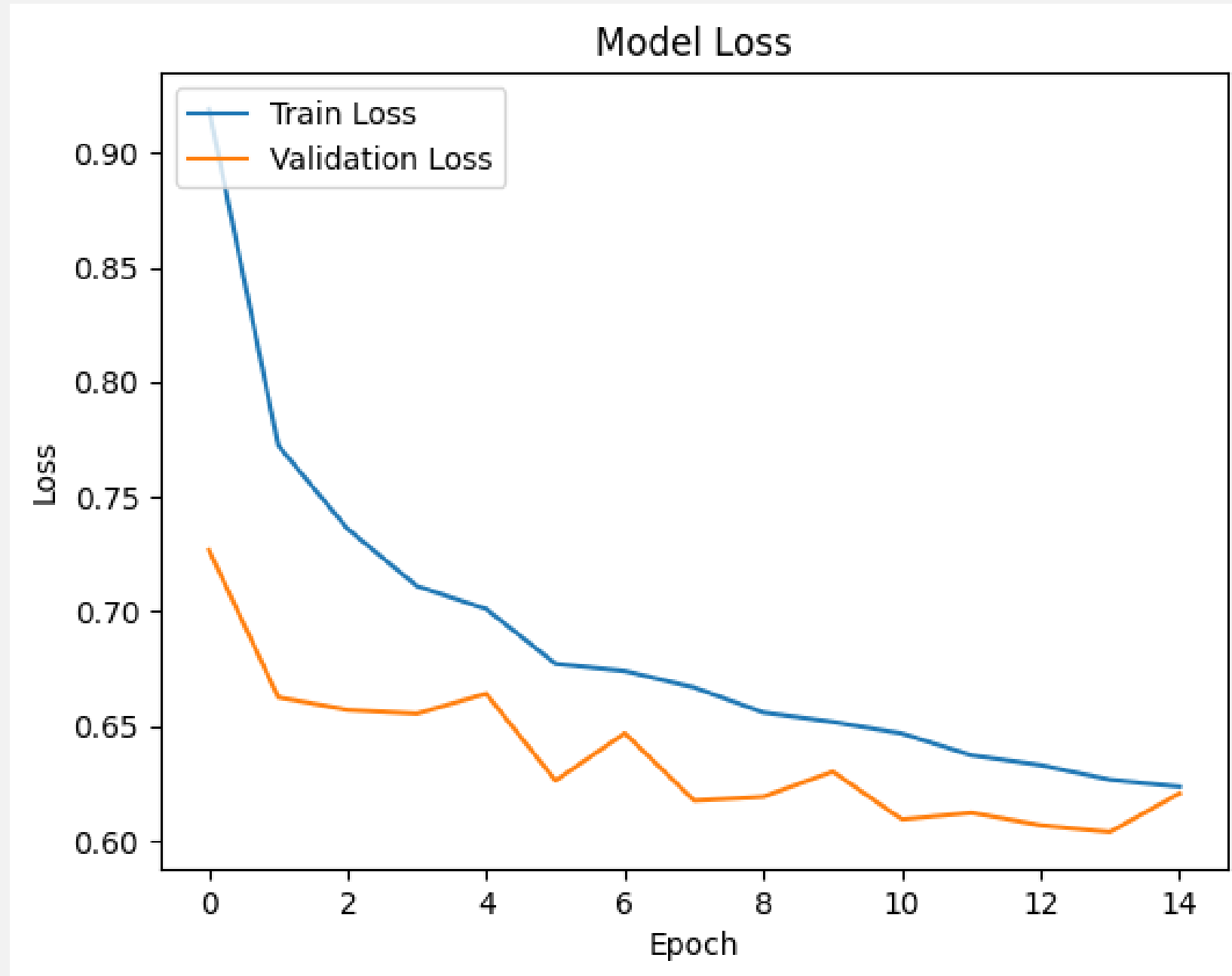


훈련 vs. 검증 손실
x축-이 축은 훈련의 반복 횟수를 나타냅니다.
y축-이 축은 손실 값(오차)를 나타냅니다.



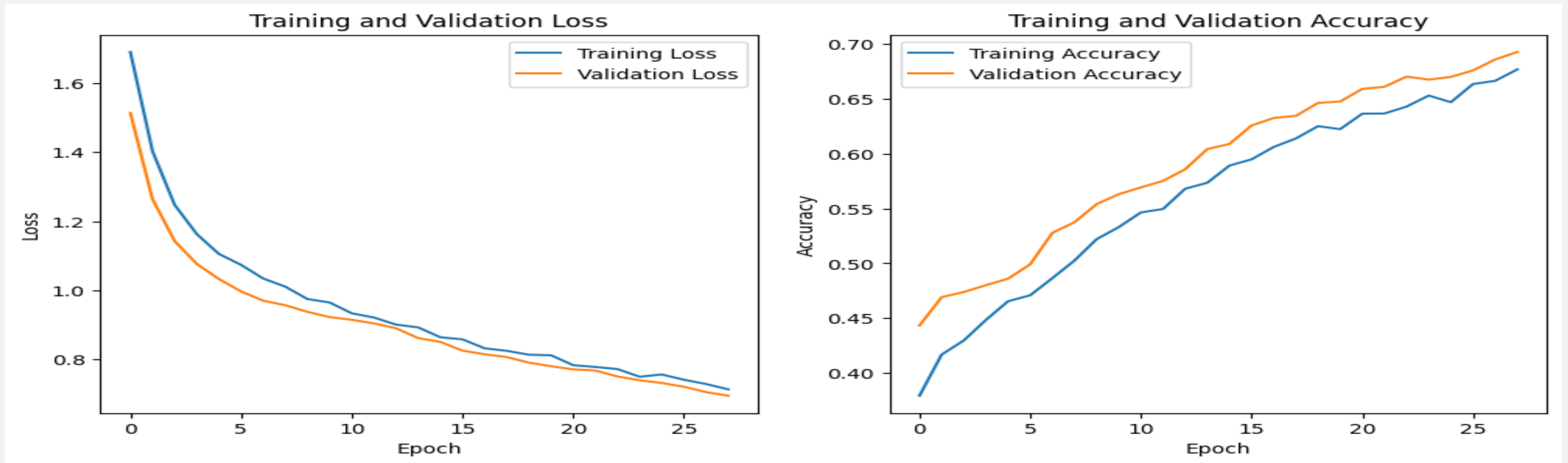
훈련 vs. 검증 정확도
x축-이 축은 훈련의 반복 횟수를 나타냅니다.
y축-이 축은 정확도를 나타냅니다.

나이 학습



이 모델은 deepface를 사용하여 사람의 얼굴 사진으로부터 나이를 예측합니다. 나이는 여섯 개의 카테고리로 분류됩니다. 15에포크 동안 훈련을 진행한 결과, 최종 테스트 정확도는 76.2%로 측정되었습니다.

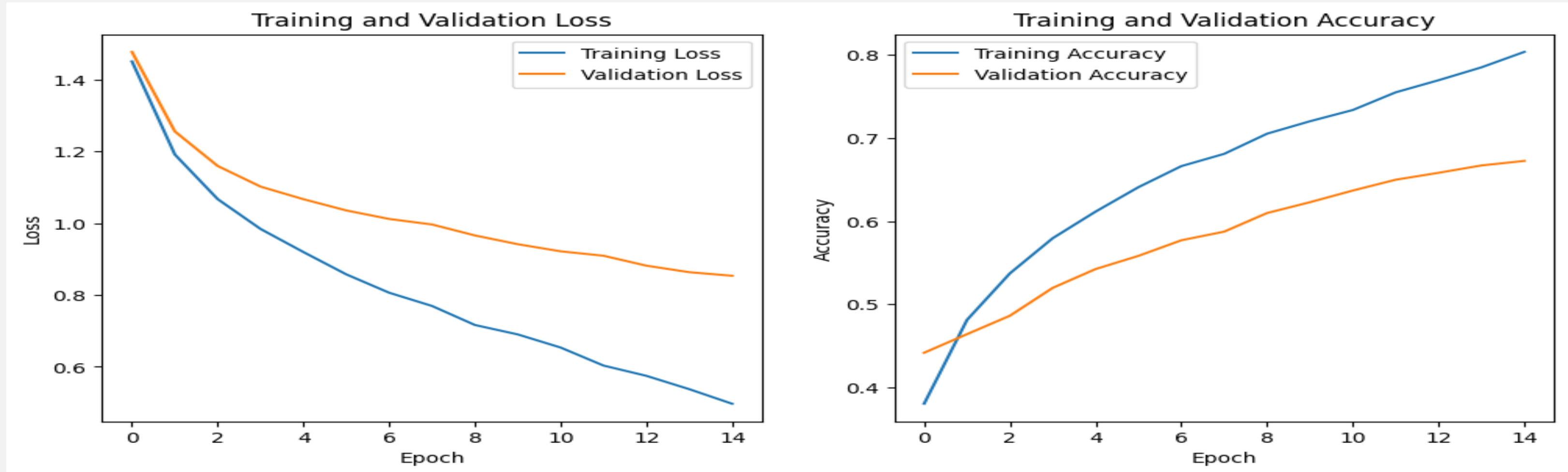
텍스트 데이터 감정 학습



- 텍스트 감정 분석

1. 모델: KcELECTRA를 기반으로 7가지 감정 분류. (기쁨, 슬픔, 화남, 중립, 놀람, 두려움, 혐오)
2. 데이터 처리: AIHub의 단발성 및 연속적 대화 데이터를 병합, 데이터 불균형 문제를 해결하기 위해 오버샘플링 기법 적용하였습니다.
3. 훈련 및 평가 결과: 모델 훈련 시 조기 종료를 사용하여 과적합을 방지했습니다. 훈련 손실과 검증 손실이 안정적으로 감소하고, 훈련 정확도와 검증 정확도가 꾸준히 증가하는 것을 볼 수 있습니다. 테스트 데이터에서 70.51%의 정확도를 기록했습니다.

음성 데이터 감정 학습

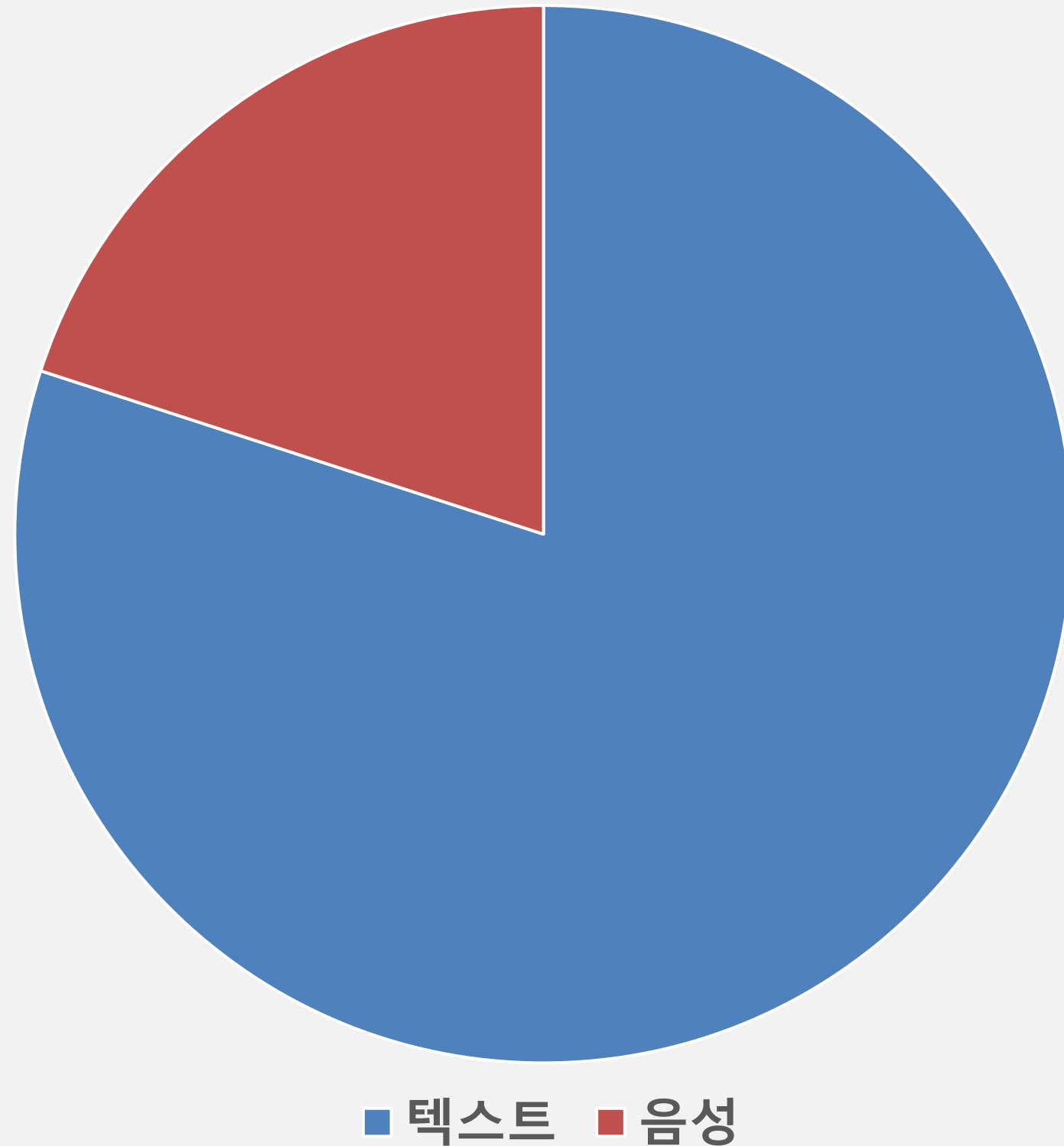


- 음성 감정 분석

1. 모델: Wav2Vec2를 기반으로 음성 데이터에서 주요 음향적 특징(피치, 볼륨, 템포)을 감정을 추출합니다.
2. 구현 : 음성 데이터를 Wav2Vec2로 처리하여 감정을 분석
3. 훈련 및 평가 결과: 모델 훈련 시 조기 종료를 사용하여 과적합을 방지했습니다. 훈련 손실과 검증 손실이 안정적으로 감소하고, 훈련 정확도와 검증 정확도가 꾸준히 증가하는 것을 볼 수 있습니다. 테스트 데이터에서 60%의 정확도를 기록했습니다.

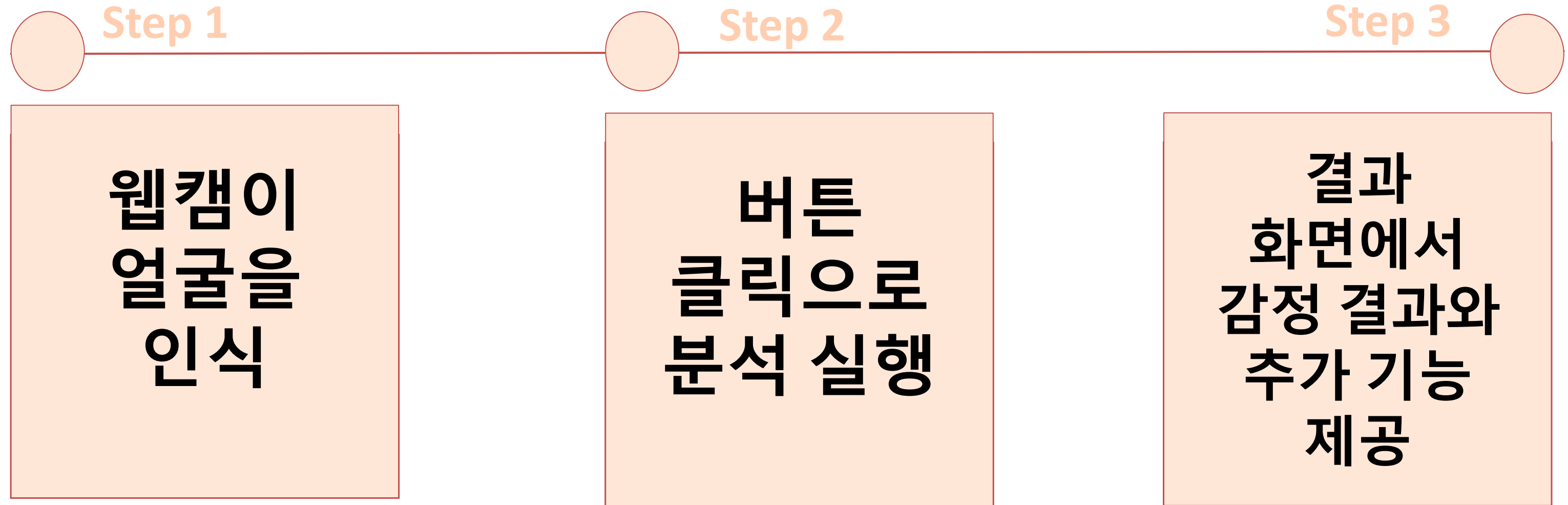
감정 모델 멀티모델 병합

가중치



- 가중치 기반 병합 (Weighted Fusion): 각 데이터 유형에 가중치를 부여하여 감정을 병합 방법입니다.
- 현재 설정: 텍스트 0.8, 음성 0.2
- 텍스트 모델의 높은 안정성을 반영하여 비중을 높게 하였습니다.
- 음성 데이터는 보조적인 역할을 하며 언어로 표현되지 않는 미묘한 감정 신호를 보완하고 보다 풍부한 감정 해석을 제공합니다
- 병합 공식
최종 감정 점수 = $0.8 \times \text{텍스트 점수} + 0.2 \times \text{음성 점수}$

UI 개발



- UI 개발 목표: PyQt5로 사용자가 실시간으로 얼굴 데이터를 분석하고 결과를 직관적으로 확인할 수 있는 환경 제공
- UI 구성
 1. 웹캠 화면: 실시간으로 얼굴을 인식하고 데이터를 캡처하는 기능. "감정 분석 시작" 버튼 클릭 시 분석 프로세스 실행
 2. 결과 화면: 분석된 성별, 나이대, 감정을 텍스트와 이모지로 표시
 3. 감정 결과에 따라 음악 선택 등 추가 작업 가능
- 작동 흐름: Step 1 웹캠이 얼굴을 인식 → Step 2 버튼 클릭으로 분석 실행 → Step 3 결과 화면에서 감정 결과와 추가 기능 제공

음악 알고리즘 및 AI 작곡

음악 알고리즘과 AI 작곡 개요



이 작품은 개인화된 음악 경험을 제공하기 위해 YouTube 음악 알고리즘, AI 음악 작곡, 그리고 가중치 학습을 결합한 시스템으로, 사용자의 나이와 감정을 기반으로 맞춤형 음악을 추천하거나 작곡해주는 서비스입니다.

이 시스템은 사용자 경험의 모든 측면에서 맞춤화와 편리함을 제공하며, 세 가지 기능의 통합을 통해 차별화된 음악 서비스를 구현합니다.

PROJECT

음악 알고리즘 사용 기술



python

주요 개발 언어로
, 모든 작곡 로직
과 MIDI 파일 생성
구현



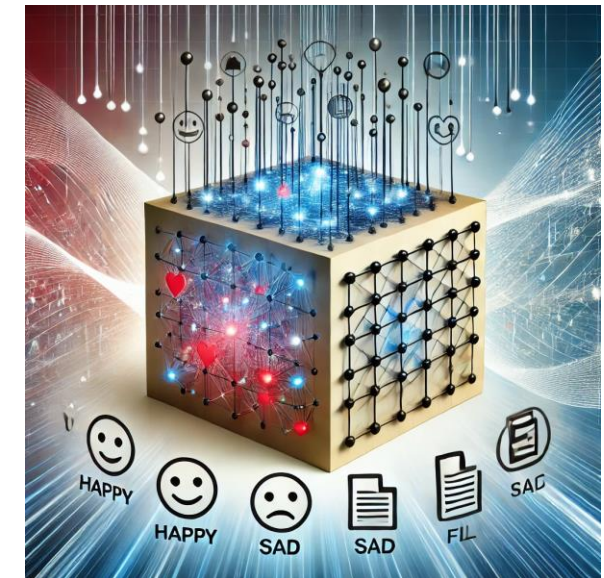
YouTube API

사용자의 나이와
감정, 자연어
처리를 반영해
음악 추천



SELENIUM

추천된 유튜브 영
상을 브라우저에
서 자동 재생



Json

사용자 피드백에
따른 가중치 데이
터를 저장 및 관리

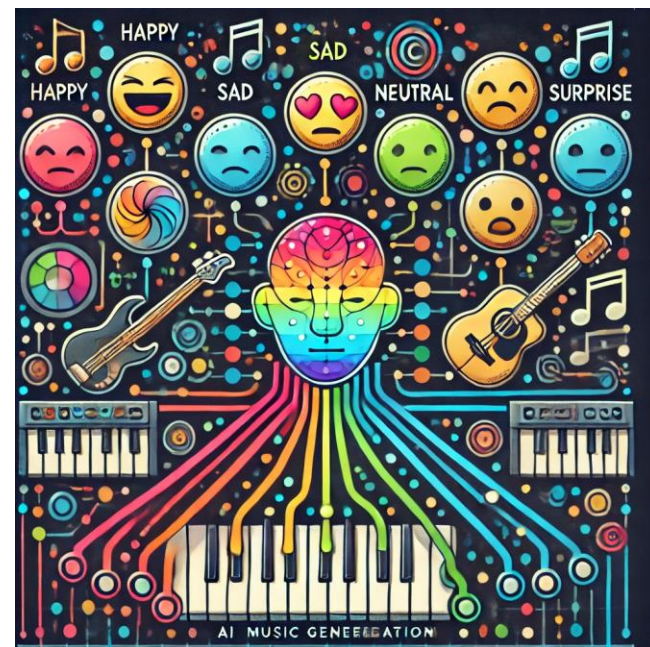
PROJECT

AI 음악 작곡 사용 기술



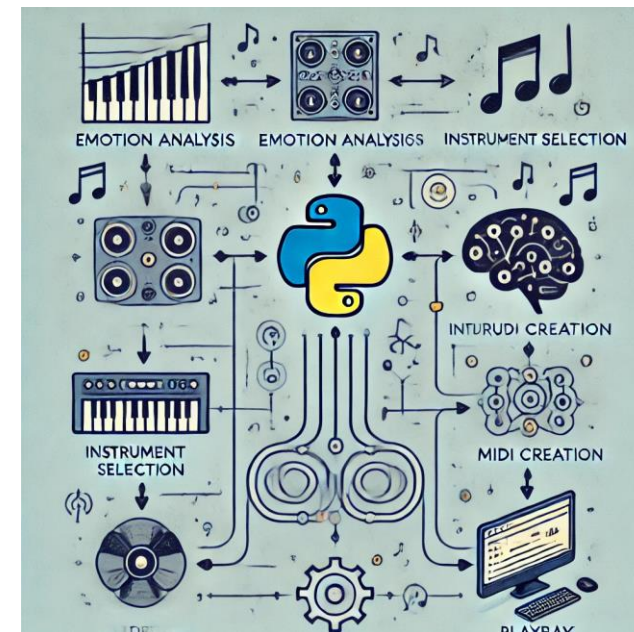
MIDIUtil
라이브러리

MIDI 파일 생성
및 작곡 자동화



Emotion-Instruments
Mapping

감정별로 어울리
는 악기, 템포, 스
케일을 매핑



멜로디 생성
알고리즘

감정에 기반한 창
의적인 코드와 멜
로디 생성



PyAudio

오디오 녹음 및
재생 지원

음악 알고리즘 구조

1단계 YouTube API

YouTube API에서 사용자의 시청 기록, 좋아요, 댓글 등을 수집하여 데이터베이스에 저장합니다.



2단계 감정 및 인구통계적 정보 분석

감정 분석 도구(자연어 처리, 얼굴 표정 분석 등)를 활용해 사용자의 감정 상태를 파악합니다. 이와 더불어 연령대와 성별에 맞춘 패턴을 감지합니다.



3단계 음악 추천 알고리즘 적용 및 가중치 학습

수집된 데이터와 감정 상태에 따라 가장 적합한 곡을 YouTube에서 실시간으로 추천합니다. 음악이 끝난 후 사용자가 생성된 음악에 대해 '좋아요/싫어요'와 같은 피드백을 제공합니다. 특정 감정에 대한 추천 가중치를 증가하거나 감소하여 다음 알고리즘에 반영합니다.



AI 음악 작곡 알고리즘 구조

1단계 감정 분석

사용자로부터 감정 데이터를 수집합니다.



2단계 음악 특징 매핑

감정에 따라 음악의 템포, 음계(스케일), 악기 등을 결정합니다.

행복: 밝은 템포(120-140 BPM), 장조 스케일, 피아노/기타.

슬픔: 느린 템포(60-80 BPM), 단조 스케일, 첼로/클라리넷.

공포: 빠른 템포(100-120 BPM), 단조 스케일, 팀파니/튜바.



3단계 멜로디와 코드 생성

MIDI 라이브러리를 사용하여 코드 노트와 멜로디 노트를 프로그램에 입력 합니다.

AI 음악 작곡 알고리즘 구조

4단계 MIDI 파일 작성

섹션별로(인트로, 벌스, 코러스, 아웃트로) 코드와 멜로디 데이터를 조합하고 MIDIFile 클래스를 사용하여 트랙, 템포, 악기 설정합니다. 최종적으로 MIDI 파일로 저장하여 사용자가 재생 가능하도록 준비합니다.



5단계 사용자 피드백 기반 학습

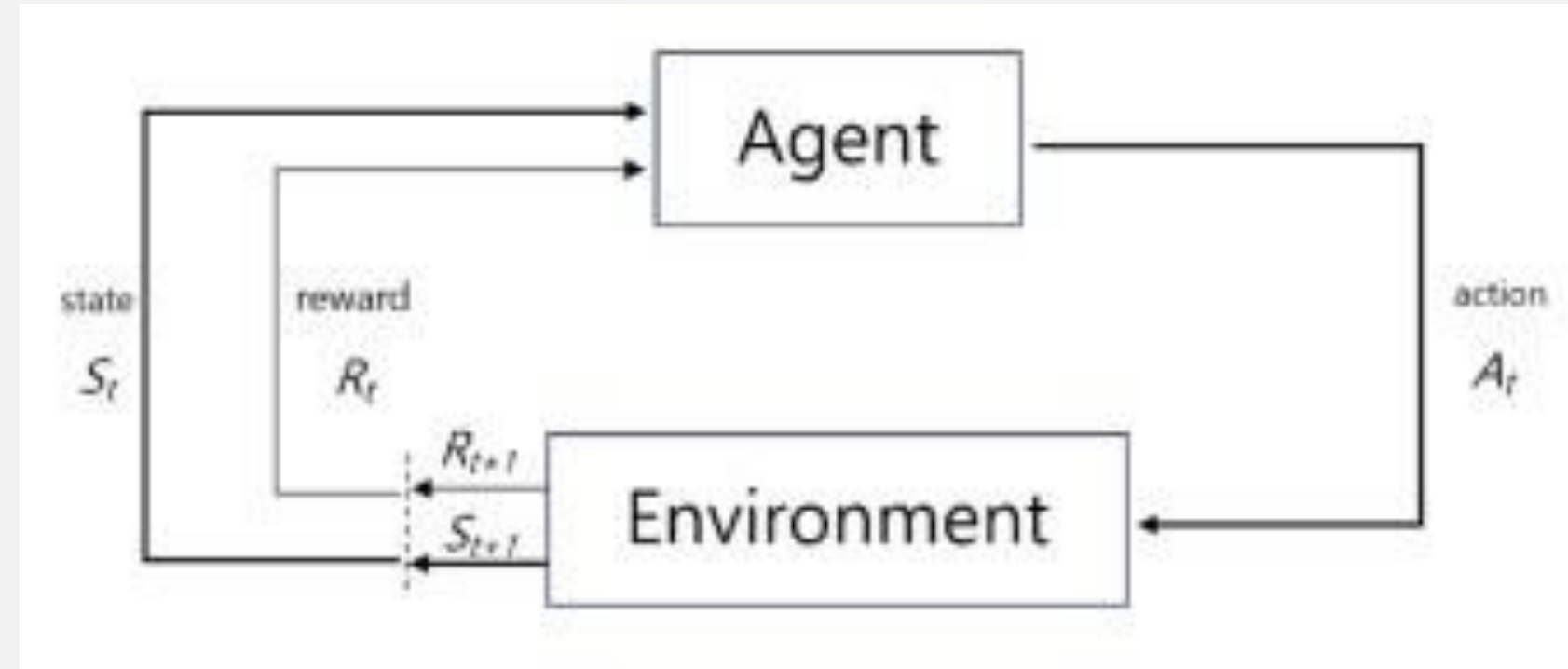
사용자가 생성된 음악에 대해 '좋아요/싫어요'와 같은 피드백을 제공합니다. 특정 감정에 대한 추천 가중치를 증가하거나 감소하여 다음 작곡에 반영.

음악 알고리즘 및 ai 음악작곡 전체 흐름도

1. 음악 추천 2. 음악작곡 및 생성

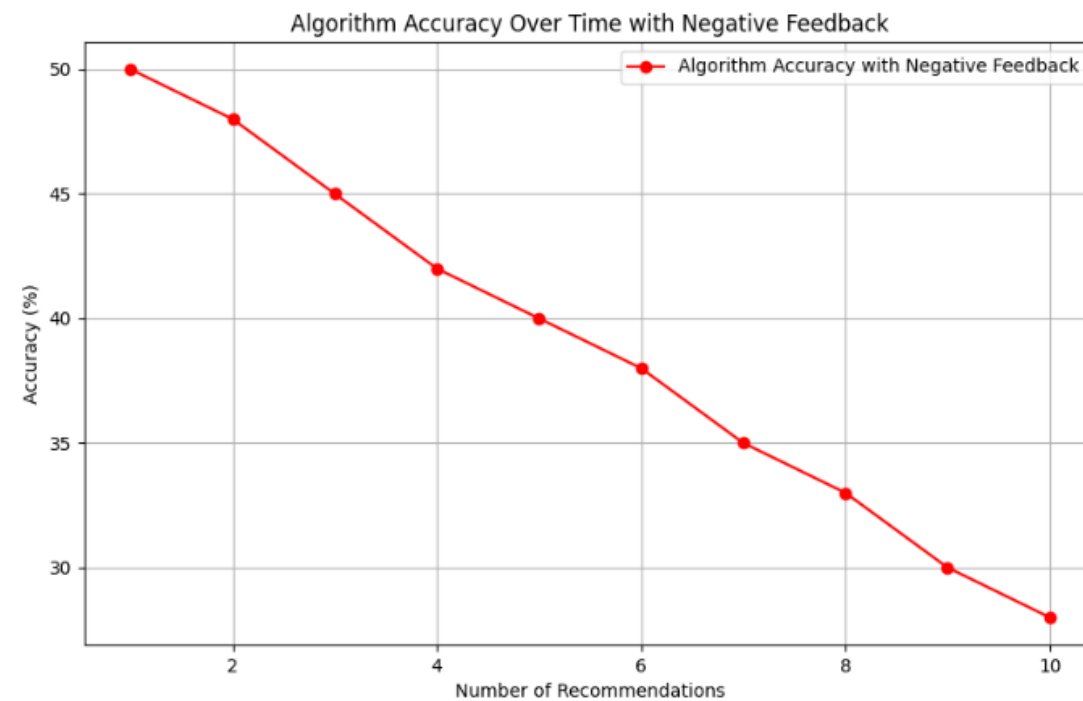
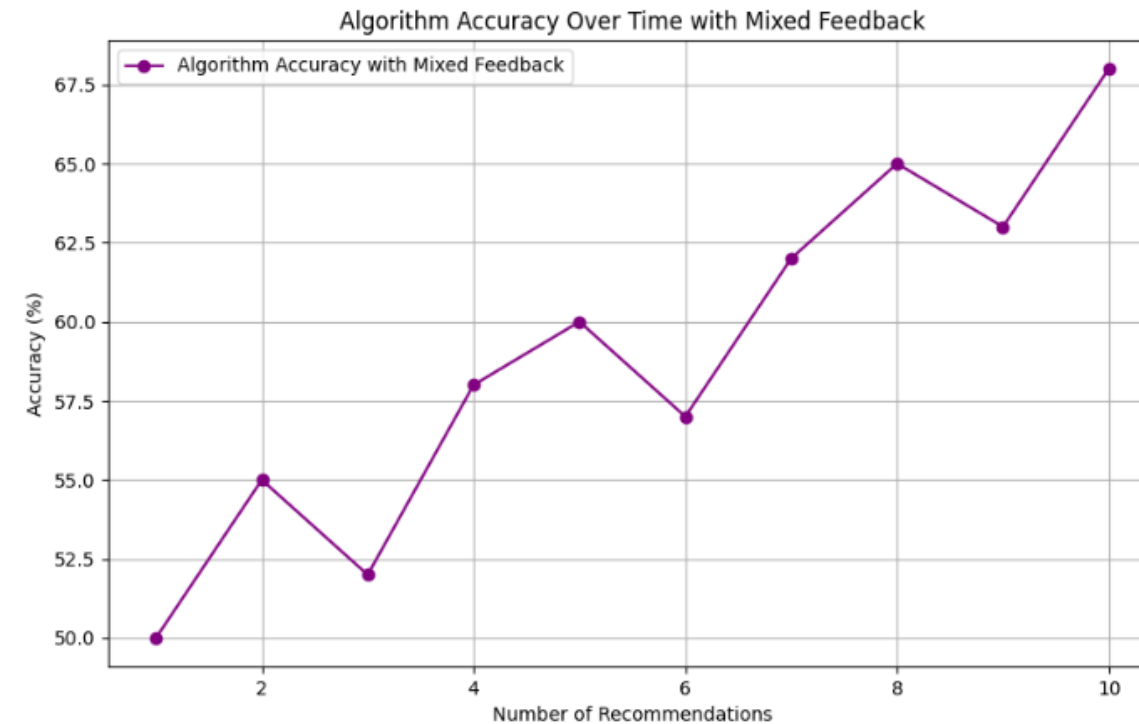
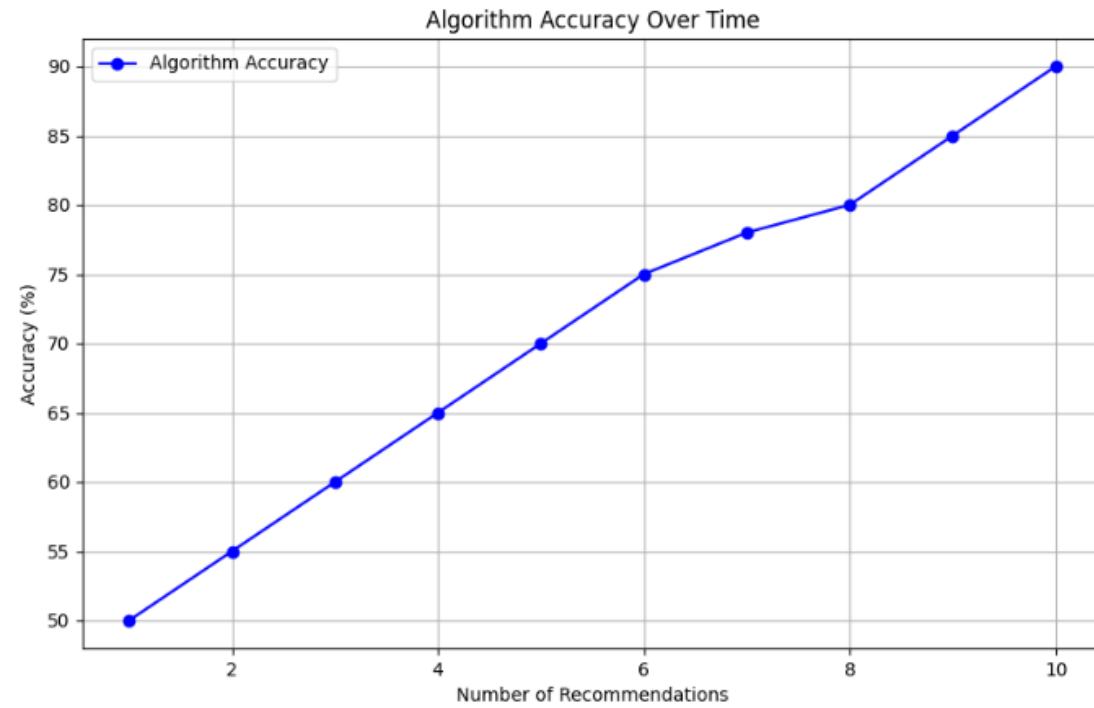


가중치 학습 설명



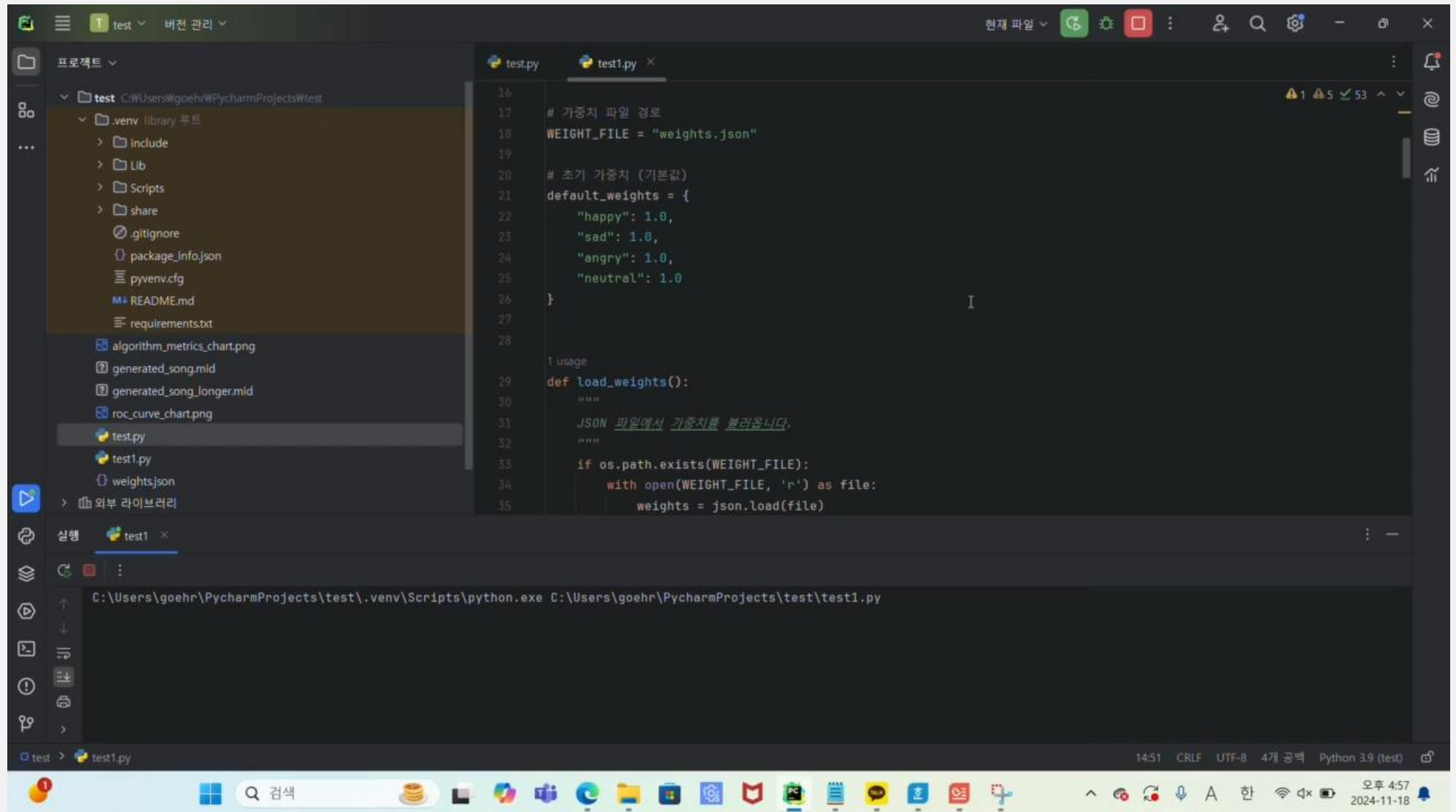
이 그림은 강화 학습의 구조를 나타내는 Agent-Environment 모델입니다. 이 구조는 저희 음악 추천 알고리즘에 적용된 핵심적인 학습 방식입니다

가중치 학습 설명



현재 알고리즘은 사용자의 감정을 기반으로 초기 추천을 제공합니다. 그러나 이는 강화 학습의 초기 상태(Exploration)이며, 이후 사용자의 피드백 데이터를 반영하여 점진적으로 맞춤형 추천을 학습(Exploitation)하도록 설계되어 있습니다. 초기 추천은 탐색(Exploration)의 일부이며, 사용자의 피드백을 통해 점진적으로 학습하여 점점 진화함 현재 나와있는표는 저희 알고리즘의 정확성을 제대로 보여주고 있습니다.

음악 추천 시연



```
16
17 # 가중치 파일 경로
18 WEIGHT_FILE = "weights.json"
19
20 # 초기 가중치 (기본값)
21 default_weights = {
22     "happy": 1.0,
23     "sad": 1.0,
24     "angry": 1.0,
25     "neutral": 1.0
26 }
27
28
29 1 usage
30 def load_weights():
31     """
32     JSON 파일에서 가중치를 불러옵니다.
33     """
34     if os.path.exists(WEIGHT_FILE):
35         with open(WEIGHT_FILE, 'r') as file:
36             weights = json.load(file)
```

실행 test1 ×

C:\Users\goehr\PycharmProjects\test\.venv\Scripts\python.exe C:\Users\goehr\PycharmProjects\test\test1.py

test test1.py 14:51 CRLF UTF-8 4개 공백 Python 3.9 (test)

음악 작곡 시연

The screenshot displays the PyCharm IDE interface. The left sidebar shows the project structure for 'test' at 'C:\Users\goehr\PycharmProjects\test'. The main editor window shows the file 'test.py' with the following code:

```
147
148
149 import openai
150 from midiutil import MIDIFile
151 import os
152
153 # 1. API 키 설정
154 openai.api_key = "sk-j3oJFS12lpIPbHCFatpls2DMk9qb5wJxz8JW6d7hZ5T3B1bkFJVirk3KYgoganYxGNa8ui27iQLsqQGkWB4g-Gb0pB4A"
155
156 # 2. MIDI 파일 생성 함수
157 1 usage
158 def create_midi_file(tempo, key, instruments):
159     # Create a new MIDI file with one track
160     midi = MIDIFile(1)
161
162     # Set up the track
163     track = 0
164     time = 0 # Start at the beginning
165     midi.addTrackName(track, time, trackName: "Generated Song")
166     midi.addTempo(track, time, tempo) # Set the tempo
```

The bottom panel shows the terminal output, indicating the execution of the script using the Python interpreter located at 'C:\Users\goehr\PycharmProjects\test\.venv\Scripts\python.exe'.

작품 시연!!!