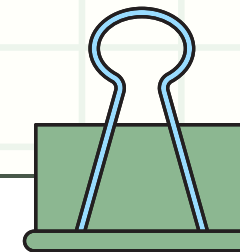


# AI를 활용하여 감정분석 웹 애플리케이션 구현

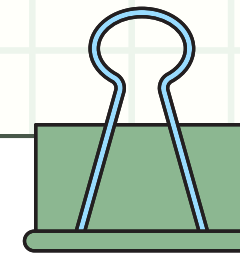
20220084 이서림, 20232531 염하나, 20240863 김주영, 20232504 이정민



# 목차

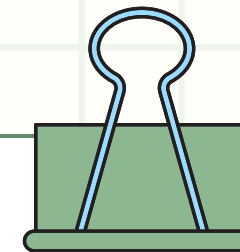
---

- 01 프로젝트 주제 소개
- 02 AI 기술활용계획  
/코드설명
- 03 프로젝트 의의



# 01 프로젝트 주제 소개

**AI기술들을 활용하여 다양한 영어 텍스트를 분석하고  
감정분석 웹 어플리케이션을 구현해본다.**



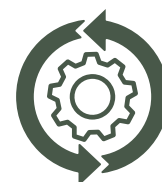
# 02 AI기술 활용계획

사용예정자료

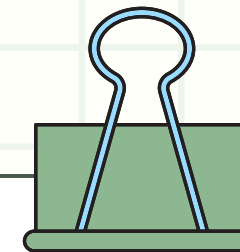


영어로 된 감정 표현 텍스트  
:미드,후기,문학,연극

활용할 AI 도구



Gradio  
Hugging face Transformers  
Open AI GPT  
Pandas



# 텍스트 예시

"Let Marmee think we are getting things for ourselves, and then surprise her."

"It's the best we've had yet," said Meg.

"You're a regular Shakespeare!" exclaimed Beth.

"Glad to find you so merry, my girls," said a cheery voice at the door.

"You're a dear, and nothing else," answered Meg warmly.

"I've got a treat for you after supper," said Mrs. March.

"A quick, bright smile went round like a streak of sunshine."

"Welcome to the real world. It sucks. You're gonna love it."

"Because you were worth the wait."

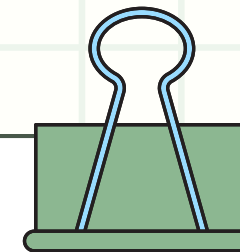
"I'm gonna miss you so much."

"We were on a break!"

"I can't believe this is over."

"I thought I'd be with someone by now. I thought I'd have a kid by now."

"I'm not good at commitment. But with you, it's different."



# AI 도구소개

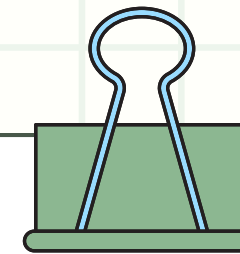
## 1. gradio

-Gradio는 머신러닝 모델, API 또는 일반 Python 함수를 위한 웹 기반 사용자 인터페이스(UI)를 쉽게 생성할 수 있도록 지원하는 오픈 소스 Python 라이브러리

### -주요 특징:

텍스트, 이미지, 오디오, 비디오 등 다양한 입력 및 출력 컴포넌트 지원  
간단한 코드로 웹 UI 생성 가능

Hugging Face Spaces와의 통합으로 모델 배포 및 공유 용이  
`gr.Interface()` 또는 `gr.Blocks()`를 사용하여 UI 구성



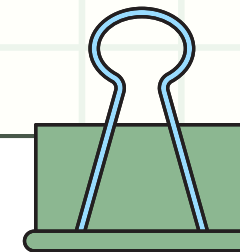
# AI 도구소개

## 2. Hugging Face Transformers 🤗

-Hugging Face Transformers는 자연어 처리(NLP), 컴퓨터 비전, 음성 인식 등 다양한 분야에서 사전 학습된 Transformer 모델을 제공하는 오픈 소스 Python 라이브러리  
PyTorch, TensorFlow, JAX와 호환되며, 복잡한 모델을 손쉽게 불러오고 사용할 수 있도록 지원

-주요 특징:

- BERT, GPT, RoBERTa, T5 등 다양한 사전 학습된 모델 제공
- pipeline API를 통해 감성 분석, 번역, 요약, 질의응답 등 다양한 작업 수행 가능
- Hugging Face Hub를 통해 모델 및 데이터셋 공유 및 다운로드 가능



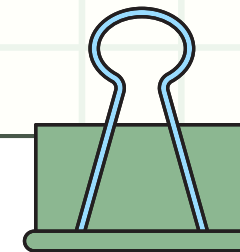
# AI 도구소개

## 3. Open AI GPT

OpenAI GPT(Generative Pre-trained Transformer)는 OpenAI에서 개발한 대규모 언어 모델로,  
인간과 유사한 자연어 생성 및 이해 능력을 갖추고 있음  
GPT-3, GPT-3.5, GPT-4 등의 버전이 있으며, 다양한 애플리케이션에서 활용되고 있음

### -주요 특징:

- 텍스트 생성, 번역, 요약, 질의응답, 코드 생성 등 다양한 작업 수행 가능
- OpenAI API를 통해 모델에 접근하여 애플리케이션에 통합 가능
- ChatGPT와 같은 대화형 AI 서비스의 기반 모델로 사용



# AI 도구소개

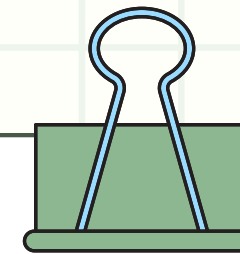
## 4.Pandas

-pandas는 Python에서 데이터 분석과 조작을 위한 오픈 소스 라이브러리로,  
특히 레이블 형태의 데이터를 효율적으로 처리할 수 있도록 지원

DataFrame과 Series라는 두 가지 주요 데이터 구조를 제공하며, 다양한 데이터 형식의 입출력 및 분석 기능을 갖추고 있음

### -주요 특징:

- CSV, Excel, SQL 등 다양한 데이터 형식의 입출력 지원
- 데이터 필터링, 그룹화, 집계, 병합 등 다양한 데이터 조작 기능 제공
- 시간 시계열 데이터 처리 및 결측치 처리 기능 지원



# 02

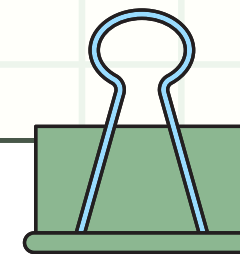
## AI기술 활용계획-코드설명

**!pip install transformers pandas**

:Python 환경에서 두 개의 주요 라이브러리인 transformers와 pandas를 설치하는 명령어

**!pip install gradio**

Python 환경에서 Gradio 라이브러리를 설치하는 명령어,  
이 명령어는 주로 Jupyter Notebook이나 Google Colab과 같은 인터랙티브 환경에서 사용



# 02

## AI기술 활용계획-코드설명

```
import pandas as pd
from transformers import pipeline
import gradio as gr
from openai import OpenAI
import random
```

```
# OpenAI API 키
client = OpenAI(api_key="sk-proj-
PmJvBAIkAkjk_8sJ34U0ncGkDEELp1QRa61a25VaQtaCrpXjnvwW8Z83c92rjtdesRJmF5o4
M2T3BlbkFJeJEifKxDS01mK2Ksw8a4MMOrAR8p5GJhFbwflMg3KAZpioeUGaqzlrXzZEXxOX
llgNaUdU_jQA") # 여기에 실제 API 키 입력
```

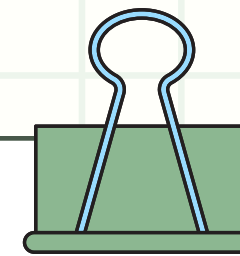
```
# 감성 분석 모델 불러오기
classifier = pipeline("sentiment-analysis")
```

```
# 파일 캐시
file_cache = {}
```

```
# GPT 해설
def gpt_explain(text, sentiment):
    prompt = f"""
문장: "{text}"
기존 감성 분류: {sentiment}
```

**사용자가 입력한 문장에 대해 감성 분석을 수행하고,  
그 결과에 대한 해설을 GPT 모델을 통해 생성하는 기능을 제공**

**gpt\_explain 함수는 번역, 감성 해설, 복합 감성 분석 등을 포함한 상세한 설명을  
반환하여, 사용자가 감성 분석 결과를 더 잘 이해할 수 있도록 도움**



# 02

## AI기술 활용계획-코드설명

다음 네 가지 항목을 제목을 포함하여 출력해주세요.

1. [번역] → 문장을 한국어로 자연스럽게 번역해 주세요.
2. [감정 분류 해설] → 해당 문장이 왜 '{sentiment}' 감정으로 분류되었는지 설명해 주세요.
3. [복합 감정 분석] → 이 문장에서 느껴지는 복합적인 감정을 분석해 주세요.  
예: 기쁨, 슬픔, 분노, 불안, 놀람, 중립 등 중 2~3개 정도
4. [복합 감정 해설] → 복합 감정 분석 결과에 대해 간단한 이유를 설명해 주세요.

각 항목 앞에는 반드시 [제목]을 붙여주세요.

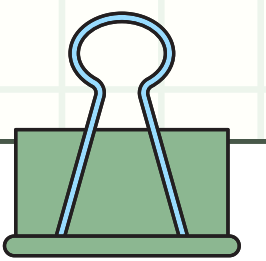
"""

```
try:
    response = client.chat.completions.create(
        model="gpt-3.5-turbo",
        messages=[{"role": "user", "content": prompt}],
        temperature=0.7,
    )
    return response.choices[0].message.content.strip()
except Exception as e:
    return f"[GPT 오류] {str(e)}"
```

1. 입력된 문장을 한국어로 자연스럽게 번역
2. 감정 분석 결과에 대한 해설을 제공
3. 문장에서 느껴지는 복합적인 감정을 분석
4. 복합 감정 분석 결과에 대한 간단한 이유를 설명

# 02

## AI기술 활용계획-코드설명



# 첫 번째 페이지 - 직접 입력 감정 분석

```
def analyze_single(text):
    result = classifier(text)[0]
    sentiment = result["label"]
    explanation = gpt_explain(text, sentiment)
    return f"감정: {sentiment} (신뢰도: {result['score']:.2f})", explanation
```

# 두 번째 페이지 - 전체 CSV 감정 분석

```
def analyze_preset_csv(type_choice):
    file_map = {
        "미드": "미드.csv",
        "후기": "후기.csv",
        "문학": "문학.csv",
        "연극": "연극.csv"
    }
```

```
    path = file_map.get(type_choice)
    if not path:
        return "유효한 선택이 아닙니다."
```

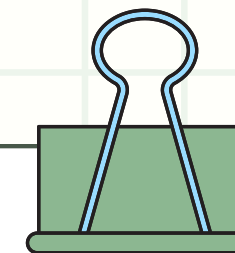
```
df = pd.read_csv(path)
if 'text' not in df.columns:
    return "'text' 컬럼이 없습니다."
```

```
df["sentiment"] = df["text"].apply(lambda x: classifier(x)[0]["label"])
df.reset_index(inplace=True)
df.rename(columns={"index": "번호"}, inplace=True)
return df
```

**첫 번째 함수는 사용자로부터 직접 입력받은 문장에 대한  
감정 분석과 해설을 제공하며,  
두 번째 함수는 선택한 카테고리의 CSV 파일 내 모든 문장에 대해  
감정 분석을 수행하고 결과를 반환**

# 02

## AI기술 활용계획-코드설명



# 세 번째 페이지 - 문장 하나씩 보기 + GPT 해설

```
def analyze_one_by_one(type_choice, idx, direction):
```

```
    file_map = {
        "미드": "미드.csv",
        "후기": "후기.csv",
        "문학": "문학.csv",
        "연극": "연극.csv"
    }
```

```
    path = file_map.get(type_choice)
```

```
    if not path:
```

```
        return "유효한 선택이 아닙니다.", "", "", idx
```

```
    if type_choice not in file_cache:
```

```
        df = pd.read_csv(path)
```

```
        file_cache[type_choice] = df
```

```
    else:
```

```
        df = file_cache[type_choice]
```

```
    if 'text' not in df.columns:
```

```
        return "'text' 컬럼이 없습니다.", "", "", idx
```

```
    if direction == "next":
```

```
        idx += 1
```

```
    elif direction == "prev":
```

```
        idx -= 1
```

```
    if idx < 0:
```

```
        return "처음입니다.", "", "", 0
```

```
    if idx >= len(df):
```

```
        return "끝에 도달했습니다.", "", "", len(df) - 1
```

```
    text = df.loc[idx, "text"]
```

```
    sentiment = classifier(text)[0]["label"]
```

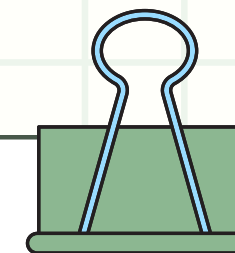
```
    explanation = gpt_explain(text, sentiment)
```

```
    return text, f"감정: {sentiment}", explanation, idx
```

**이 함수는 사용자가 선택한 카테고리의 CSV 파일에서  
문장을 하나씩 순차적으로 불러와 감정 분석을 수행하고,  
그 결과에 대한 GPT 해설을 제공  
파일 캐싱을 통해 성능을 향상시키며,  
인덱스 범위를 검사하여 사용자에게 적절한 메시지를 제공  
이러한 기능은 Gradio 인터페이스의 "문장 하나씩 살펴보기" 탭에서 활용**

# 02

## AI기술 활용계획-코드설명



# 네 번째 페이지 - 감정 퀴즈

```
def get_quiz_question(category):
```

```
    file_map = {
```

```
        "미드": "미드.csv",
```

```
        "후기": "후기.csv",
```

```
        "문학": "문학.csv",
```

```
        "연극": "연극.csv"
```

```
    }
```

```
    path = file_map.get(category)
```

```
    if not path:
```

```
        return "유효하지 않은 선택입니다.", "", "", 0
```

```
    df = pd.read_csv(path)
```

```
    idx = random.randint(0, len(df) - 1)
```

```
    text = df.loc[idx, "text"]
```

```
    label = classifier(text)[0]["label"]
```

```
    return text, label, idx
```

# 퀴즈 정답 체크 + GPT 해설 추가하기

```
def check_answer(user_choice, correct_label, text):
```

```
    if user_choice.upper() == correct_label:
```

```
        result = "정답입니다! 🎉"
```

```
    else:
```

```
        result = f"틀렸어요! 정답은 {correct_label}입니다."
```

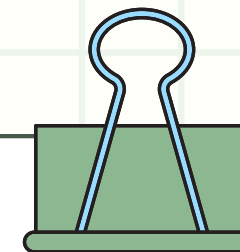
```
        explanation = gpt_explain(text, correct_label)
```

```
    return result, explanation
```

**get\_quiz\_question** 함수는 선택한 카테고리의 CSV 파일에서 무작위로 문장을 선택하고, 해당 문장의 감정을 분석하여 퀴즈 문제를 생성

**check\_answer** 함수는 사용자의 선택과 정답을 비교하여 결과를 판단하고, GPT를 통해 해설을 제공

이러한 구조를 통해 사용자는 다양한 문장을 기반으로 감정 분석 퀴즈를 풀고, GPT의 해설을 통해 학습할 수 있음



# 02 AI기술 활용계획-코드설명

# Gradio UI 구성

with gr.Blocks() as demo:

```
gr.Markdown("<h2 style='text-align: center;'>감정 분석 앱</h2>")
```

# 1. 문장 직접 입력

with gr.Tab("1. 문장 직접 입력"):

```
input_text = gr.Textbox(label="문장 입력", lines=4)
```

```
output_label = gr.Textbox(label="감정 분석 결과")
```

```
gpt_output = gr.Textbox(label="GPT 해설", lines=8)
```

with gr.Row():

```
clear_btn = gr.Button("Clear", variant="secondary")
```

```
submit_btn = gr.Button("Submit", variant="primary")
```

```
submit_btn.click(analyze_single, inputs=input_text, outputs=[output_label, gpt_output])
```

```
clear_btn.click(lambda: ["", "", ""], outputs=[input_text, output_label, gpt_output])
```

# 2. 전체 CSV 분석

with gr.Tab("2. 전체 데이터 감정 분석"):

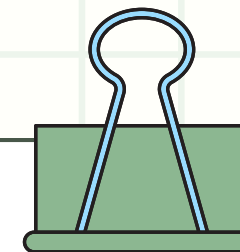
```
dropdown = gr.Radio(choices=["미드", "후기", "문학", "연극"], label="카테고리 선택")
```

```
start_btn = gr.Button("선택한 CSV 분석")
```

```
output_df = gr.DataFrame()
```

```
start_btn.click(analyze_preset_csv, inputs=dropdown, outputs=output_df)
```

이 Python 코드는 감정 분석 웹 애플리케이션을 구현한 것으로,  
사용자가 문장을 입력하면 해당 문장의 감정을 분석하고,  
추가로 GPT-3.5를 활용하여 해설을 제공하는 기능을 갖추고 있음  
또한, CSV 파일을 통한 대량 감정 분석, 문장별 감정 분석,  
감정 퀴즈 등의 기능도 포함되어 있음



# 02

## AI기술 활용계획-코드설명

### # 3. 문장 하나씩 보기

```
with gr.Tab("3. 문장 하나씩 살펴보기"):
    category = gr.Radio(choices=["미드", "후기", "문학", "연극"], label="카테고리 선택")
    sentence_output = gr.Textbox(label="문장", lines=3)
    sentiment_output = gr.Textbox(label="감정 결과")
    explanation_output = gr.Textbox(label="GPT 해설", lines=8)
    index_state = gr.State(0)
    with gr.Row():
        prev_btn = gr.Button("⏮ 이전 문장")
        next_btn = gr.Button("다음 문장 ⏭")
    next_btn.click(
        analyze_one_by_one,
        inputs=[category, index_state, gr.Textbox(value="next", visible=False)],
        outputs=[sentence_output, sentiment_output, explanation_output, index_state]
    )
    prev_btn.click(
        analyze_one_by_one,
        inputs=[category, index_state, gr.Textbox(value="prev", visible=False)],
        outputs=[sentence_output, sentiment_output, explanation_output, index_state]
    )
```

### # 4. 감정 퀴즈

```
with gr.Tab("4. 감정 퀴즈 맞히기"):
    gr.Markdown("문장을 보고 감정이 긍정적인지 부정인지 맞혀보세요!")
    quiz_category = gr.Radio(choices=["미드", "후기", "문학", "연극"], label="카테고리 선택")
    quiz_question = gr.Textbox(label="문제 문장", lines=3, interactive=False)
    quiz_choices = gr.Radio(choices=["POSITIVE", "NEGATIVE"], label="감정 선택")
    quiz_result = gr.Textbox(label="결과", interactive=False)
    quiz_explanation = gr.Textbox(label="GPT 해설", lines=8)
    correct_label_state = gr.State("")
    quiz_idx = gr.State(0)
```

```
get_btn = gr.Button("문제 가져오기")
submit_btn = gr.Button("제출하기")
```

```
get_btn.click(get_quiz_question, inputs=quiz_category, outputs=[quiz_question, correct_label_state, quiz_idx])
submit_btn.click(
    check_answer,
    inputs=[quiz_choices, correct_label_state, quiz_question],
    outputs=[quiz_result, quiz_explanation]
)
```

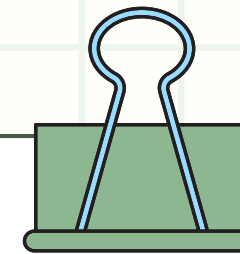
### # 실행

```
demo.launch(share=True)
```

**사용자는 문장을 입력하거나 CSV 파일을 선택하여  
감정 분석을 수행하고, 결과에 대한 GPT 기반 해설을 받을 수 있음**

- 1.문장 직접 입력 탭
2. 전체 CSV 데이터 감정 분석 탭
- 3.문장 하나씩 살펴보기 탭
- 4.감정 퀴즈 맞히기 탭
- 5.실행

**이 과정을 통해 다양한 방식으로 감정분석체험,  
결과에 대해 심층적인 해설을 받을수있음**



## 감정 분석 앱

1. 문장 직접 입력   2. 전체 데이터 감정 분석   3. 문장 하나씩 살펴보기   4. 감정 퀴즈 맞추기

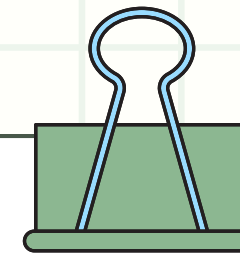
문장 입력

감정 분석 결과

GPT 해설

Clear

Submit



## 감정 분석 앱

1. 문장 직접 입력   2. 전체 데이터 감정 분석   3. 문장 하나씩 살펴보기   4. 감정 퀴즈 맞추기

### 문장 입력

I think maybe I'm not such a people person anymore. I've missed you so much.

### 감정 분석 결과

감정: NEGATIVE (신뢰도: 0.64)

### GPT 해설

#### [번역]

"아마 나는 이제 사람을 좋아하는 사람이 아닌 것 같아. 너를 너무나 그리워했어."

#### [감정 분류 해설]

이 문장은 "I think maybe I'm not such a people person anymore." 부분에서 자신이 사람을 좋아하지 않는다고 느끼고 있으며, "I've missed you so much." 부분에서 그리워하고 있는 상대방을 향한 감정을 표현하고 있어서 NEGATIVE 감정으로 분류됩니다.

#### [복합 감정 분석]

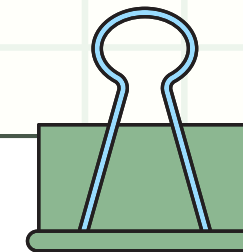
- 보고 싶음 (그리움)
- 자기 인식의 변화 (불안)

#### [복합 감정 해설]

이 문장은 상대방을 그리워하면서도 자신의 사람을 좋아하지 않는다는 생각에 불안함을 느끼는 복합적인 감정을 담고 있습니다. 상대방을 그리워하는 감정과 자신의 변화에 대한 불안함이 교차하는 모습이 보입니다.

Clear

Submit



## 감정 분석 앱

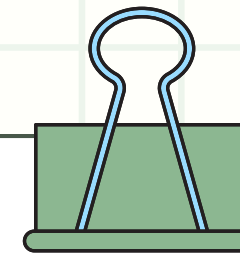
1. 문장 직접 입력    2. 전체 데이터 감정 분석    3. 문장 하나씩 살펴보기    4. 감정 퀴즈 맞추기

카테고리 선택

☒ 미드    ☐ 후기    ☐ 문학    ☐ 연극

### 선택한 CSV 분석

| 번호 | text                                                                     | sentiment |
|----|--------------------------------------------------------------------------|-----------|
| 0  | "It's always been you, Rach."                                            | POSITIVE  |
| 1  | "Welcome to the real world. It sucks. You're gonna love it."             | POSITIVE  |
| 2  | "Because you were worth the wait."                                       | POSITIVE  |
| 3  | "I'm gonna miss you so much."                                            | POSITIVE  |
| 4  | "We were on a break!"                                                    | POSITIVE  |
| 5  | "I can't believe this is over."                                          | NEGATIVE  |
| 6  | "I thought I'd be with someone by now. I thought I'd have a kid by now." | NEGATIVE  |
| 7  | "I'm not good at commitment. But with you, it's different."              | POSITIVE  |
| 8  | "I never thought I'd be able to love someone so much."                   | POSITIVE  |
| 9  | "He's not dead! I can feel it."                                          | POSITIVE  |
| 10 | "I dumped your ass."                                                     | NEGATIVE  |
| 11 | "You don't have to do this alone."                                       | POSITIVE  |



## 감정 분석 앱

1. 문장 직접 입력    2. 전체 데이터 감정 분석    3. 문장 하나씩 살펴보기    4. 감정 퀴즈 맞추기

카테고리 선택

☒ 미드    ☐ 후기    ☐ 문학    ☐ 연극

문장

"Welcome to the real world. It sucks. You're gonna love it."

감정 결과

감정: POSITIVE

GPT 해설

[번역]  
한국어: "진짜 세상에 오신 걸 환영합니다. 현실은 참으로 싱겁지만, 당신은 이를 즐기게 될 거예요."

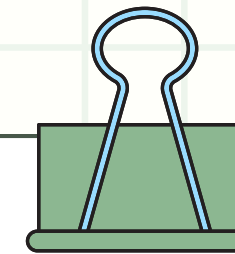
[감정 분류 해설]  
이 문장은 처음에는 부정적인 내용을 담고 있지만, 그것을 긍정적인 방향으로 바라보며 받아들이라는 메시지를 전달하고 있기 때문에 'POSITIVE' 감정으로 분류된 것으로 보입니다.

[복합 감정 분석]  
- 현실에 대한 실망  
- 긍정적인 전망

[복합 감정 해설]  
이 문장은 현실이 어렵고 싱겁다는 부정적인 감정을 담고 있지만, 동시에 그것을 받아들이고 긍정적으로 바라보라는 메시지가 담겨 있어서 혼합된 감정을 느끼게 됩니다. 이러한 복합적인 감정이 'POSITIVE'로 분류된 이유입니다.

[← 이전 문장](#)

[다음 문장 →](#)



## 감정 분석 앱

1. 문장 직접 입력    2. 전체 데이터 감정 분석    3. 문장 하나씩 살펴보기    4. 감정 퀴즈 맞추기

문장을 보고 감정이 긍정인지 부정인지 맞춰보세요!

카테고리 선택

☐ 미드

☐ 후기

☐ 문학

☒ 연극

문제 문장

(Reads) "April 3rd to April 5th: Three days of rain." (She looks up or him.) That's all?

감정 선택

☐ POSITIVE

☒ NEGATIVE

결과

정답입니다! 🎉

GPT 해설

[번역]

"(읽으면서) '4월 3일부터 4월 5일까지: 세 일 동안 비!' (그는 그를 향해 시선을 돌린다.) 그게 다였어요?"

[감정 분류 해설]

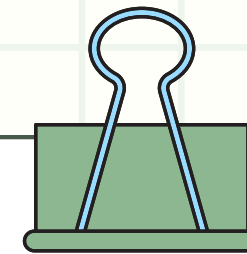
이 문장은 간단하게 세 일 동안 계속되는 비에 대한 불만과 실망이 담겨 있어서 'NEGATIVE' 감정으로 분류될 수 있습니다.

[복합 감정 분석]

실망, 지루함

[복합 감정 해설]

이 문장은 단순히 세 일 동안 비가 온다는 사실에 대한 지루함과 함께 그것으로 인해 기대에 못 미치는 실망을 느끼게 합니다.



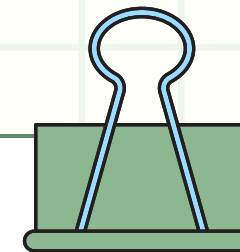
# 프로젝트 의의

1. 사용자가 감정을 예측해 본 뒤, AI의 분석 및 해설과 비교하며 자신의 해석을 점검하고 사고를 확장할 수 있습니다.

2. 다양한 학습 방식(문장 입력, CSV 분석, 퀴즈 등)을 통해 능동적인 감정 분석 경험이 가능합니다.

3. 단순한 감정 분류를 넘어, 감정이 그렇게 분류된 이유와 뉘앙스를 이해할 수 있도록 돕는 것에 의의가 있습니다.

4. 미드, 후기, 문학 등 다양한 장르의 문장을 통해 감정 표현의 폭넓은 예시를 경험할 수 있습니다.



감사합니다

THANK  
YOU!