상담사 5 대 지표 평가 시스템

모델 정의서

상담사 성과 평가 및 자동 피드백 생성 시스템

팀원: 오현서, 김기훈, 노준석, 오정우

목차

1. 프로젝트 개요

1.1 프로젝트 주제 및 배경

- 주제: 상담사 성과 평가 및 자동 피드백 생성

- 부제: 5 대 핵심 지표 기반 상담사 평가 알고리즘과 OpenAI API 연동 피드백 시스템

- 프로젝트 의의: 상담사의 5 가지 핵심 역량을 정량적으로 평가하고 맞춤형 코칭 피드백을 자동 생성하여 상담 품질 향상과 교육 효율성을 극대화하는 것을 목표로 함. 실제 평가 알고리즘과 AI 기반 자연어 피드백을 결합한 완전 자동화 평가 파이프라인을 제시

- 배경: 상담사 평가 방식 선정 근거

- 일반적으로 상담사 평가 방식은 주관적 관찰 평가 vs 정량적 데이터 기반 평가 vs AI 기반 자동 평가로 구분할 수 있음
- 정량적 데이터 기반 평가 방식은 객관적이고 일관된 기준으로 대규모 상담사를 효율적으로 평가할 수 있으며, 최근 자연어처리와 감정분석 기술의 발달로 더욱 주목받고 있음
- 특히 정량적 평가 방식은 상담 대화록이나 고객 만족도 데이터 같은 기존 수집 데이터를 활용할 수 있어 별도 평가 인프라 구축 없이도 도입할 수 있다는 장점이 있음
- 평가 결과가 수치로 저장되기 때문에 상담사뿐만 아니라 관리자도 직관적으로 성과를 파악할 수 있으며, AI 피드백을 통해 구체적 개선방안까지 제시 가능
- 정량적 평가의 한계점인 맥락 이해 부족은 추후 대화 맥락 분석과 연동하여 보완 가능 (비용 및 시간적 한계)

1.2 프로젝트 목적

- 평가 대상 지표 리스트

- 정중함 및 언어 품질: 존댓말 사용률, 긍정적 언어 표현, 부정적 언어 표현, 완곡하는 언어 표현
- 공감적 소통: 고객 감정 이해 및 적절한 공감 표현
- 문제해결 역량: 구체적 해결책 제시 능력
- 감정 안정성: 상담 과정에서 고객 감정 개선 유도
- 대화 흐름 및 응대 태도: 원활한 대화 진행과 적절한 응답 속도

- 목적

- 실제 평가 알고리즘을 통한 정확한 점수 산출: absolute_grading 시스템을 활용하여 5 대 지표별로 정확하고 일관된 점수와 등급을 산출함. 백분위 기반 A~G 등급 체계로 객관적 성과 측정이 가능함
- OpenAI GPT 기반 전문적 피드백 자동 생성: 산출된 점수를 바탕으로 상담사 교육 전문가 역할의 GPT 가 강점/약점 분석 및 구체적 코칭 멘트를 자동 생성함. 개별 상담사별 맞춤형 개선 전략 제공이 가능함
- 완전 자동화된 평가 파이프라인 구축: 데이터 입력부터 최종 피드백 생성까지 완전 자동화하여 대규모 상담사 평가와 실시간 성과 모니터링을 지원함. Supabase 연동으로 평가 결과의 체계적 관리와 추적이 가능함

2. 데이터 정의 및 처리

2.1. 기준 데이터셋 (dummy_data.csv)

- **역할**: 전체 평가 시스템의 일관성과 객관성을 담보하는 **'평가 기준선(Baseline)'을 수립**하는 데 사용되는 표준 데이터.
- **구성**: session_id 를 포함하여 5 대 지표를 계산하는 데 필요한 모든 정량적 Feature 컬럼들로 구성됨.

2.2. 신규 평가 데이터 (new_data.csv)

- **역할**: 시스템이 실제로 평가를 수행해야 할 대상 데이터.
- **구성**: 기준 데이터셋과 동일한 형식의 컬럼을 가짐.

2.3. 데이터 전처리 (이상치 처리)

- **목적**: 데이터의 안정성과 평가 결과의 신뢰도를 높이기 위해 극단적인 값을 보정.
- **방식**: absolute_grading 폴더 내 스크립트 실행 시, **IQR(사분위수 범위) 기법**을 사용하여 각 Feature 의 극단적인 값(Outlier)을 자동으로 감지하고 경계값으로 조정함.

```python
def clip_outliers_iqr(df, cols):
    for col in cols:
        q1 = df[col].quantile(0.25)
        q3 = df[col].quantile(0.75)
        iqr = q3 - q1
        lower = q1 - 1.5 * iqr
        upper = q3 + 1.5 * iqr
        df[col] = df[col].clip(lower, upper)
    return df
```

3. 시스템 아키텍처 및 평가로직

3.1. 시스템 구성

본 시스템은 전통적인 머신러닝 '모델 학습' 과정을 거치지 않음. 대신, 사전 분석된 데이터(dummy_data.csv)를 기반으로 수립된 '평가 기준선'을 사용하여 새로운 데이터를 평가하는 규칙 기반 시스템임.

- 평가 기준선 (Baseline): cutoff 폴더 내 JSON 파일들. 각 평가지표의 등급(A~G)을 나누는 점수(Cut-off)와 데이터 정규화(Normalization)를 위한 최소/최대값(Min/Max)을 정의.
- 평가 알고리즘: absolute_grading 폴더 내 Python 스크립트들. 새로운 데이터(new_data.csv)가 들어오면, 평가 기준선을 바탕으로 점수와 등급을 계산.
- LLM 피드백 생성: LLM_evaluation_batch.py 스크립트. 평가 알고리즘이 산출한 점수/등급을 프롬프트에 담아 OpenAI GPT API 에 전달하고, 맞춤형 피드백을 생성.

3.2. 평가 기준선 수립 (calculate_cutoff.py)

- 목적: 평가의 일관성과 객관성을 담보하는 기준점을 설정.
- 프로세스:
    1. 기준 데이터 로드: data/dummy_data.csv 파일을 로드.
    2. 점수 일괄 계산: 5 대 지표별 평가 알고리즘을 실행하여 모든 기준 데이터의 점수를 계산.
    3. 백분위 기반 Cut-off 산출: 각 지표별 점수 분포의 백분위(90%, 80%, 70% 등)를 계산하여 A~G 등급을 나누는 경계 점수를 결정.
    4. Min/Max 값 산출: 데이터 정규화에 사용될 각 Feature 의 최소값과 최대값을 계산.

5. 기준선 저장: 산출된 Cut-off 와 Min/Max 값을 cutoff 폴더 내에 지표별 JSON 파일로 저장.

```jsonc
// from cutoff/grade_cutoff_empathy.json
{
  "cutoff": {
    "A": 0.789, "B": 0.712, "C": 0.630,
    "D": 0.561, "E": 0.492, "F": 0.427,
    "G": -1000000000.0
  },
  "minmax": {
    "empathy_ratio": { "min": 0.0, "max": 30.0 },
    "apology_ratio": { "min": 0.0, "max": 5.0 }
  }
}
```

3.3. 신규 데이터 평가 프로세스 (absolute_grading/)

1. **신규 데이터 로드**: data/new_data.csv 파일을 로드.
2. **이상치 처리 (IQR)**: 데이터의 안정성을 위해 사분위수 범위를 이용해 극단적인 값(Outlier)을 보정.
3. **기준선 범위 검증**: 신규 데이터의 Min/Max 값이 기존 cutoff 파일에 저장된 Min/Max 범위를 벗어나는지 확인.
    - **범위 내일 경우**: 기존 기준선을 그대로 사용하여 평가를 진행.
    - **범위를 벗어날 경우**: dummy_data.csv 와 new_data.csv 를 통합하여 새로운 기준선을 동적으로 재산출하고, 이를 cutoff 폴더에 업데이트.
4. **점수 및 등급 산출**: 업데이트된 기준선을 바탕으로 각 상담 세션의 최종 점수와 등급을 계산.

```python
# from absolute_grading/grade_empathy_auto.py
# ... (데이터 및 기준선 로드) ...

# 새로운 데이터의 min/max가 기존 범위를 벗어나는지 체크
if check_minmax(new_minmax, old_minmax):
    print('기존 cut-off/minmax로 평가')
    minmax = old_minmax
else:
    print('범위 벗어남 → cut-off/minmax 재산출')
    # ... (dummy_data와 합쳐서 기준선 재산출 로직) ...

# 최종 점수 및 등급 산출
for col in cols:
    eval_df[f'{col}_norm'] = minmax_normalize(eval_df[col], minmax[col]['min'], minmax[col]
['max'])
eval_df['Empathy_score'] = eval_df.apply(compute_empathy_score, axis=1)
eval_df['Empathy_Grade'] = eval_df['Empathy_score'].apply(lambda x: grade_from_cutoff(x, cutoffs)
```

4. 평가 알고리즘 상세

4.1. 5 대 평가 지표 및 계산 공식

평가 지표	측정 요소	계산 공식 (요약)
정중함 및 언어 품질	존댓말, 긍정/부정어, 완곡어 사용 비율	4 개 Feature 의 정규화 점수를 가중 평균
공감적 소통	공감 및 사과 표현 비율	2 개 Feature 의 정규화 점수를 가중 평균 (공감 70%, 사과 30%)
문제 해결 역량	구체적 해결책 제시 수준	사전 정의된 4 단계 절대 점수 매핑
감정 안정성	상담 전후 고객 감정 변화	초기 감정, 최종 감정, 개선도를 복합적으로 분석
대화 흐름 및 응대 태도	대화 끊김, 침묵 비율, 발화 균형	3 개 Feature 의 정규화 점수를 가중 평균 (침묵 40%)

```python
# from absolute_grading/grade_stability_auto.py
def compute_stability_score(row):
    ic_norm = row['interruption_count_norm']
    sr_norm = row['silence_ratio_norm']
    tr_norm = row['talk_ratio_norm']

    interrupt_score = 1 - ic_norm

    optimal_silence = 0.25
    silence_distance = abs(sr_norm - optimal_silence)
    silence_score = max(0.0, 1 - 4 * silence_distance)

    talk_distance = abs(tr_norm - 0.5)
    talk_score = max(0.0, 1 - 2 * talk_distance)

    score = interrupt_score * 0.3 + silence_score * 0.4 + talk_score * 0.3
    return float(np.clip(score, 0.0, 1.0))
```

4.2. 등급 체계

- **백분위 기반 7 단계 등급 (A~G)**: '문제 해결 역량'을 제외한 4 개 지표에 적용. 기준 데이터셋의 점수 분포에 따라 상위 10%는 A, 20%는 B 와 같이 상대적인 등급을 부여.
- **절대 기준 4 단계 등급 (A, B, C, D)**: '문제 해결 역량' 지표에만 적용. 해결 수준(완전 해결=1.0)에 따라 절대적인 등급을 부여.

5. 기술 사양 및 실행 방법

5.1. 사용 기술 및 라이브러리

- **주요 기술**: Min-Max 정규화, IQR 이상치 처리, 백분위 기반 등급 산정, OpenAI GPT API 연동
- **라이브러리**: pandas, numpy, openai, python-dotenv
- **데이터베이스 (선택적 연동)**: supabase

5.2. 메인 실행 방법

1. **환경 변수 설정**: 프로젝트 루트에 .env 파일을 생성하고 OpenAI API 키를 입력.
2. **평가 데이터 준비**: data/ 폴더에 평가할 new_data.csv 파일을 위치시킴. (없을 경우 dummy_data.csv 로 실행됨)
3. **메인 스크립트 실행**:
4. **실행 결과**: 터미널에 각 상담 세션별 5 대 지표 점수/등급 및 LLM 이 생성한 최종 피드백이 출력됨.

```
# from LLM_evaluation_batch.py
# ... (평가 결과(evaluation_result) 계산 후) ...

# OpenAI 프롬프트 생성
prompt = f"""
당신은 상담사 교육 전문가입니다. 아래 5가지 지표 평가 결과를 바탕으로...

📊 **평가 결과**
- 정중함: {evaluation_result['Politeness']['score']:.3f}점 ({evaluation_result['Politeness']['grade']}등급)
- 공감: {evaluation_result['Empathy']['score']:.3f}점 ({evaluation_result['Empathy']['grade']}등급)
...
"""
try:
    response = client.chat.completions.create(
        model=MODEL_NAME,
        messages=[{"role": "user", "content": prompt}],
        # ...
    )
    feedback = response.choices[0].message.content
except Exception as e:
    feedback = f"OpenAI API 호출 실패: {e}"
```

6. 향후 개선 및 발전 방향

- 코드 리팩토링: LLM_evaluation_batch.py 와 LLM_evaluation_with_supabase.py 등 유사 스크립트를 단일화하고, 설정 파일(config)을 통해 DB 연동 및 모델 선택을 제어하도록 개선.

- 테스트 코드 도입: pytest 등을 활용한 단위 테스트를 추가하여 각 평가 로직의 안정성 확보.
- 이벤트 기반 아키텍처: (DB 연동 시) 단순 폴링 방식 대신, Supabase 웹훅 또는 DB 함수를 활용하여 데이터 입력 시 자동으로 평가가 트리거되는 이벤트 기반 시스템으로 고도화.
- 대화 맥락 분석: 단순 키워드 비율을 넘어, 문맥을 이해하는 Transformer 기반의 자연어 처리 모델을 도입하여 평가 정확도 향상.

7. 부록

- 상세 알고리즘 설명

정중함 및 언어 품질 평가 (Politeness)

수식: (존댓말_비율 + 긍정어_비율 + 완곡어_비율 + (1 - 부정어_비율)) / 4

- 특징: 4 개 언어 품질 지표의 균등 가중평균
- 정규화: Min-Max (0~1 범위)
- 등급: A~G (백분위 기반)

공감적 소통 평가 (Empathy)

수식: 공감표현_비율 × 0.7 + 사과표현_비율 × 0.3

- 특징: 공감 표현에 더 높은 가중치 부여
- 정규화: Min-Max (0~1 범위)
- 등급: A~G (백분위 기반)

문제해결 역량 평가 (Problem Solving)

매핑: 완전해결(1.0)→A, 대부분해결(0.6)→B, 부분해결(0.2)→C, 미흡(0.0)→D

- 특징: 이산형 절대 점수 체계
- 정규화: 불필요 (사전 정의된 값)
- 등급: A~D (4 단계)

감정 안정성 평가 (Emotional Stability)

수식: 복합적 감정 변화 분석

if 감정변화 == 0:
    점수 = 초기감정 수준에 따른 보정값
else:
    점수 = 후기감정 × 0.7 + 개선도 × 0.3

- 특징: 감정 변화량과 최종 상태 모두 고려
- 정규화: Min-Max (0~1 범위)
- 등급: A~G (백분위 기반)

대화 흐름 및 응대 태도 평가 (Stability)

수식: 중단점수 × 0.3 + 침묵점수 × 0.4 + 발화균형점수 × 0.3

- 중단점수: 1 - 정규화된_중단횟수
- 침묵점수: 1 - 4 × |정규화된_침묵비율 - 0.25| (0.25 가 최적)
- 발화균형점수: 1 - 2 × |정규화된_발화비율 - 0.5| (0.5 가 최적)
- 특징: 침묵 비율에 가장 높은 가중치 부여
- 정규화: Min-Max (0~1 범위)
- 등급: A~G (백분위 기반)

OpenAI GPT 피드백 시스템

- 역할 설정: 상담사 교육 전문가
- 출력 형식: 강점 분석 + 약점 분석 + 실행 가능한 코칭 멘트
- 개인화: 5 대 지표 점수를 바탕으로 맞춤형 피드백 생성
- 품질 보장: 구체적이고 실행 가능한 개선방안 제시