

여행지 리뷰/인기도 분석 웹앱 상세 기획안

프로젝트명: **Travel Insight+**

목표: 다양한 플랫폼의 여행 리뷰를 통합 분석해 인기 여행지, 계절별 트렌드, 리뷰 키워드를 시각적으로 제공하는 웹앱

1. 데이터 수집 (스크래핑) 상세 전략

1.1 대상 플랫폼별 스크래핑 포인트

(1) 트립어드바이저 (TripAdvisor)

- 대상 URL: <https://www.tripadvisor.com/Attractions>
- 수집 데이터:
 - 여행지명, 위치(위도/경도), 카테고리(명소, 음식점, 숙소)
 - 평균 별점(1~5점), 리뷰 수, 리뷰 텍스트, 작성 일자
 - "방문 목적"(커플/가족/단체) 태그 (있는 경우)
- 도구: **Scrapy** + **Rotating Proxies** (차단 회피)
- 주의점: **Pagination** 처리, **CAPTCHA** 우회 필요

(2) 네이버 여행 (Naver Travel)

- 대상 URL: <https://travel.naver.com>
- 수집 데이터:
 - 국내 여행지 한정 리뷰 (강릉, 제주도 등)
 - 해시태그(#감성카페 #가족여행), 평점, 방문 월별 데이터
- 도구: **Selenium** (동적 렌더링 대응)
- 특이사항: 로그인 우회 필요 (간헐적 팝업 발생)

(3) 에어비앤비 (Airbnb)

- 대상 URL: `https://www.airbnb.com/rooms/`
 - 수집 데이터:
 - 숙소별 리뷰 텍스트, 별점, 위치 기반 주변 명소
 - "청결도" "접근성" 등 세부 평가 항목
 - 도구: 공식 API (혹은 `Puppeteer`로 HTML 파싱)
 - 한계: API Rate Limit (분당 100회 제한)
-

2. 데이터 분석 상세 방법론

2.1 인기 여행지 순위 알고리즘

python

Copy

가중치 공식 (예시)

```
def calculate_rank(avg_rating, review_count, recent_reviews):
```

```
    weight_rating = avg_rating * 0.6
```

```
    weight_popularity = min(review_count / 1000, 1.0) * 0.3 # 리뷰 수 보정
```

```
    weight_recency = (recent_reviews / review_count) * 0.1 # 최신성 반영
```

```
    return weight_rating + weight_popularity + weight_recency
```

- 시각화: 지도에 **Heatmap + Top 10 Bar Chart** (D3.js)

2.2 계절별 수요 분석

- 데이터 전처리:
 - 리뷰 작성일 → 계절(봄/여름/가을/겨울) 매핑
 - "12월" → "겨울", "7월" → "여름" 변환
- 분석 방법:
 - 계절별 리뷰 수 비교 → Pie Chart

- **Word2Vec**으로 계절 연관 키워드 추출 (예: "겨울" → "스키", "온천")

2.3 리뷰 요약 & 감성 분석

(1) 키워드 추출

- **TF-IDF + Noun Extraction (KoNLPy)**

- python
- Copy

```
from konlpy.tag import Okt
```

```
okt = Okt()
```

- `nouns = okt.nouns(review_text)` # 명사만 추출
- 시각화: WordCloud (색상으로 긍정/부정 구분)

(2) 감성 분석

- 모델: **KoBERT** (한글), **VADER** (영어)
- 출력 예시:

Copy

[긍정 리뷰 75%]

👍 "경관이 아름답고 음식이 맛있어요" (4.5점)

- 🙄 "주차장이 협소해서 불편했음" (2.0점)

3. 웹앱 기능별 상세 설계

3.1 메인 페이지 (/)

- 컴포넌트:
 1. 실시간 인기 여행지 **TOP 5** (Carousel 슬라이드)
 2. 검색 바 (지역/키워드 자동완성)

3. 계절 추천 섹션 (예: "이번 여름엔 제주도 해수욕장!")

3.2 검색 결과 페이지 (/search?q=강릉)

- 레이아웃:

```
html
Copy
```

<왼쪽 사이드바>

- 필터: 별점(4점 이상), 계절(여름), 방문 목적(가족)

<메인 영역>

- 여행지 카드 (이미지 + 별점 + 키워드 요약)

- 지도 미니맵 (Google Maps API)

```
Run HTML
```

3.3 상세 분석 페이지 (/detail/강릉안목해변)

- 데이터 시각화:
 - 리뷰 트렌드 그래프: 최근 1년간 월별 리뷰 수 변화
 - 감성 분석 도넛 차트: 긍정/부정 비율
 - 키워드 네트워크: 연관어 클러스터링 (Gephi 활용)

4. 기술 스택 상세

분야	기술	선택 이유
프론트엔드	React + TypeScript	컴포넌트 재사용성 높음

백엔드	FastAPI (Python)	비동기 크롤링에 최적화
데이터베이스	MongoDB	비정형 리뷰 데이터 저장 적합
분석 모델	KoBERT, Scikit-learn	한국어 NLP 지원
시각화	D3.js, Chart.js	인터랙티브 그래프 구현

5. 개발 프로세스 (Agile Sprint 2주 단위)

Sprint 1: 데이터 수집 인프라 구축

- 트립어드바이저 스크래퍼 완성 (+Proxy 설정)
- MongoDB Schema 설계 (여행지, 리뷰 컬렉션)

Sprint 2: 분석 모델 개발

- 한국어/영어 감성 분석 파이프라인 구축
- TF-IDF 키워드 추출 테스트

Sprint 3: 웹앱 MVP 출시

- 검색 → 결과 페이지 연결
- 지도 API 연동 (카카오맵)

Sprint 4: 고도화

- 사용자 피드백 반영 (예: "겨울 여행지 필터 추가")
- 크롤링 스케줄링 (매주 자동 갱신)

6. 리스크 관리

리스크	대응책
-----	-----

크롤링 차단	User-Agent 로테이션, 요청 간 딜레이 추가
--------	------------------------------

데이터 부족	구글 리뷰 백업 소스 확보
--------	----------------

모델 정확도 저하	수동 라벨링 데이터 추가 학습
-----------	------------------

최종 결과물 예시 (스크린샷)

1. 메인 페이지
2. 상세 분석 페이지

다음 단계:

1. 스크래핑 **PoC**: 트립어드바이저에서 제주도 10개 명소 데이터 수집 테스트
2. **Figma** 디자인: [링크 템플릿](#) 참고해 UI 초안 작성
3. **API** 명세서 작성: `/api/search?q=강릉` 샘플 응답값 설계

필요한 부분이 있으면 언제든지 질문해주세요! 😊