

Python과 Streamlit을 활용한 정시 합격 예측 시스템

데이터 기반의 대학 추천 알고리즘 구현

조선대학교 경영학부 20251569 김승연

2025년 11월 23일

기획 배경 (Why?)

- **문제 의식**

- 수능 직후, 복잡한 점수 계산과 수많은 대학 리스트로 인한 수험생들의 혼란
- 기존 입시 사이트의 복잡한 UI와 유료 서비스의 접근성 문제

- **목표**

- "내 점수만 입력하면, 갈 수 있는 대학을 직관적으로 보여주자"
- Python을 활용한 빠르고 간편한 웹 서비스 프로토타입 개발

- **Language: Python 3.9**
 - 데이터 처리 및 조건 논리 구현 용이
- **Web Framework: Streamlit**
 - 별도의 HTML/CSS 없이 Python만으로 빠른 UI 구축 가능
- **Version Control: Git & GitHub**
 - 코드 버전 관리 및 Streamlit Cloud 연동 배포
- **Library: Pandas**
 - 데이터 구조화 및 추후 확장성 고려

핵심 알고리즘 1: 데이터 구조 및 처리

- **데이터 구조 (Dictionary)**

- 대학명(Key)과 커트라인(Value)을 맵핑하여 빠른 검색 가능
- 예: {"S": 390, "K": 382 ...}

- **입력 데이터 처리**

- 국어, 수학, 영어, 탐구1, 탐구2 점수의 합산 (Summation)
- $\text{Total_Score} = \text{Kor} + \text{Math} + \text{Eng} + \text{Tam1} + \text{Tam2}$

핵심 알고리즘 2: 구간 분류 로직

- 합격 가능성 판단 로직 (**If-Elif Structure**)
 - 내 점수와 커트라인의 차이(`diff`)를 계산

1. **안정 (Safe)**: $diff \geq 10$
(커트라인보다 10점 이상 여유)
2. **적정 (Match)**: $-5 \leq diff < 10$
(± 5 점 내외의 접전 구간)
3. **소신/상향 (Reach)**: $-15 \leq diff < -5$
(추가 합격을 노리는 구간)

- 사용자 친화적 인터페이스

- st.number_input: 정확한 숫자 입력을 위한 위젯 사용
- st.tabs: 정보를 한 번에 나열하지 않고, 탭으로 구분하여 가독성 확보
- st.spinner: 분석 중 로딩 애니메이션을 추가하여 사용자 경험(UX) 향상

- [실제 웹 사이트 구동 화면 시연]
- 시나리오
 1. 점수 입력 (평균적인 점수)
 2. 분석 버튼 클릭
 3. '적정' 탭 결과 확인
 4. 점수 수정 (고득점) 후 '안정' 탭 변화 확인

한계점 및 향후 발전 과제

- **현재의 한계**

- 원점수 단순 합산 방식 (표준점수, 백분위 미반영)
- 하드코딩된 적은 데이터 샘플

- **향후 계획 (Future Works)**

1. **공공 데이터 포털 API 연동:** 실제 작년도 입시 결과 데이터 확보
2. **가중치 적용:** 대학별 과목 반영 비율 계산 로직 추가
3. **시각화 강화:** 내 위치를 그래프로 보여주는 기능 추가

감사합니다.

Q & A