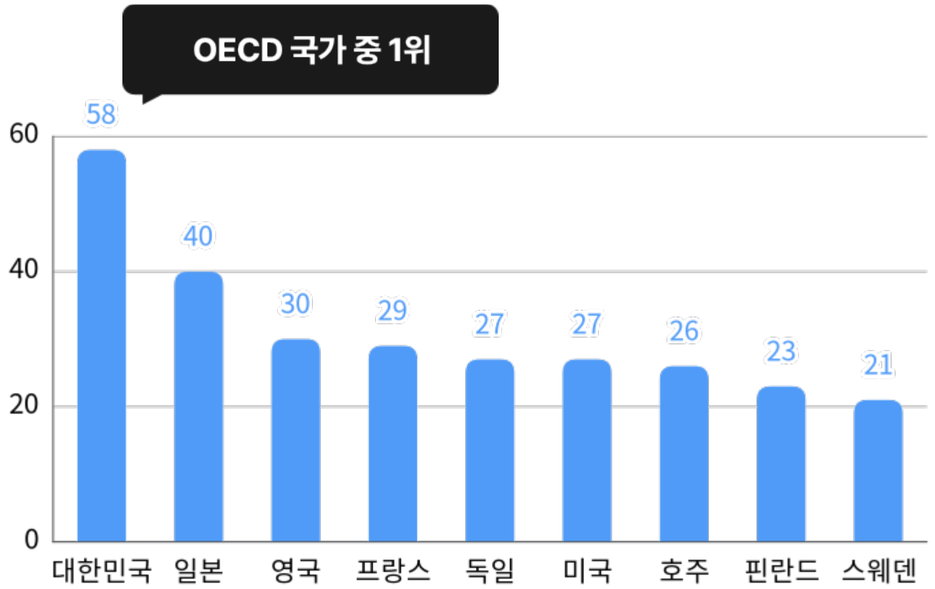
도보 속도 예측을 통한 개인 맞춤형 출발 시점 추천 시스템

데이터 엔지니어링 Team 5
변민우, 박주연, 김유진

01 문제상황

OECD 국가별 평균 통근 소요 시간 (편도)



* 국토교통부 및 한국교통연구원

직장인 평균 출근길 소요 시간

서울 거주 **55분**

경기인천 거주 **78분**

78분 x 2 x 260일

= 연간 약 28일

* 잡코리아 / 남녀직장인 277명 대상 설문

01 문제상황

왜 아무도 능동적으로 개선할 생각을 하지 않았을까?

기존 교통 앱

- 수동적 정보성 기능에 집중

도보 소요 시간
대중교통 소요 시간
대중교통 정류소 도착 예정 시간

제시하는 기능

- 능동적 서비스 기능에 집중

대중교통 이용 시간에서 줄일 수 있는 시간을 최대한 줄여주는 서비스

01 문제상황

대중교통 이용에 소요되는 시간이 긴 이유

+ 교통체증

✓ 정류장에 빨리 도착해 대중교통을 기다리는 시간

+ 거리

✓ 대중교통을 놓쳐서 다음 배차를 기다리는 시간

✓ 대중교통을 놓치지 않기 위해 잡는 여유 시간

+ 정류소 위치

01 문제상황

대중교통 정류소까지 가는 도보시간

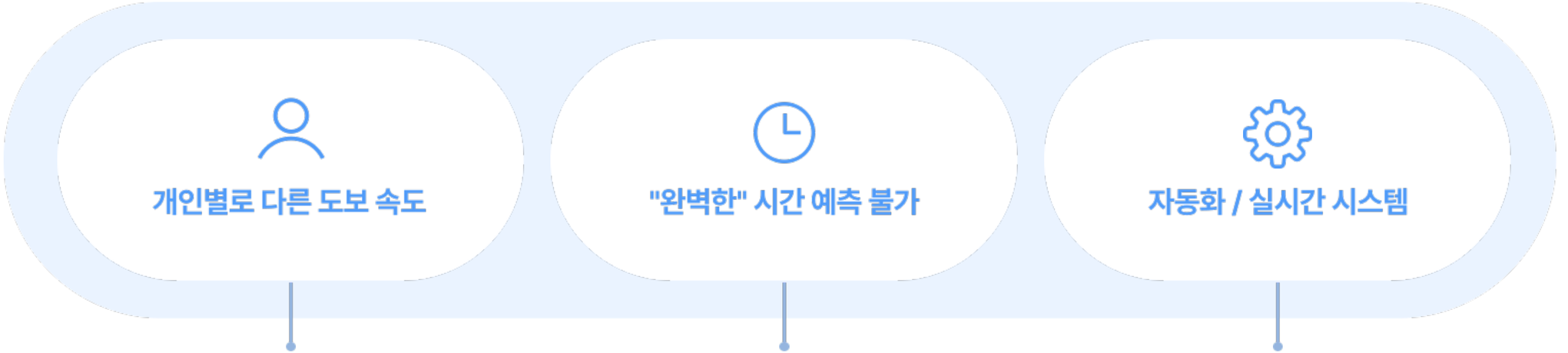


너무 빨리 또는 늦게 정류소에 도착해
정류소에서 기다리는 시간 최소화

02 프로젝트 목표

사용자 개개인에게 **맞춤형 도보 시간 예측 시스템** 제공

최소한의 데이터로 최대한의 최적화



피드백 루프를 통한
예측 모델 파인 튜닝

제시간에 도착할
확률별 출발 시간 제공

수집하는 데이터 **최소화**
데이터 처리 **정밀화**

03 결과

핵심 기능

데이터 기반 예측

실시간 버스 위치 및 사용자 보행 데이터를 활용한
정확한 도착 시간 예측



개인 맞춤화

사용자별 보행 속도 및 시간 보정 계수를 적용한
맞춤형 예측



확률 기반 의사결정

50% / 95% / 99%
제시간에 정류소에 도착할 확률별
출발 시점 추천



지속적 학습

피드백 루프를 통한 모델 자동 개선

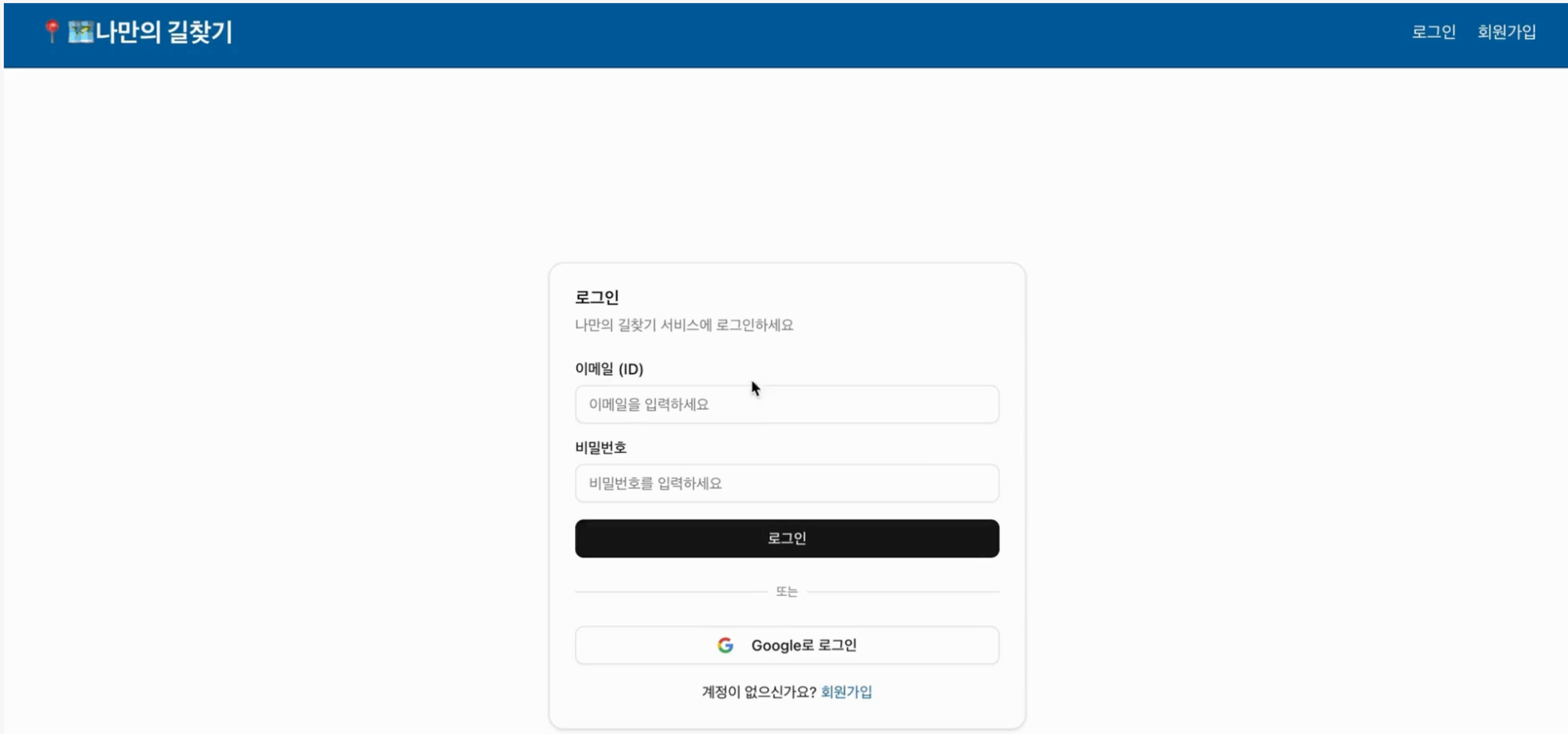


03 결과

서비스 웹 링크

*현재 서울 및 몇몇 주요 광역시만 사용 가능

*폰에서 사용 시 가로모드 사용



https://data-engineering-omega.vercel.app/login

04 데이터 파이프라인



데이터 수집

- 사용자 위치
- 버스 위치
- 보행 속도

데이터 저장

-PostgreSQL
- 예측 로그
- 피드백 데이터

데이터 처리

- 보정 계수 계산
- 통계 분석
- 확률 계산

예측 모델

- 도보 시간
- 버스 도착 시간
- 버스를 놓칠 확률

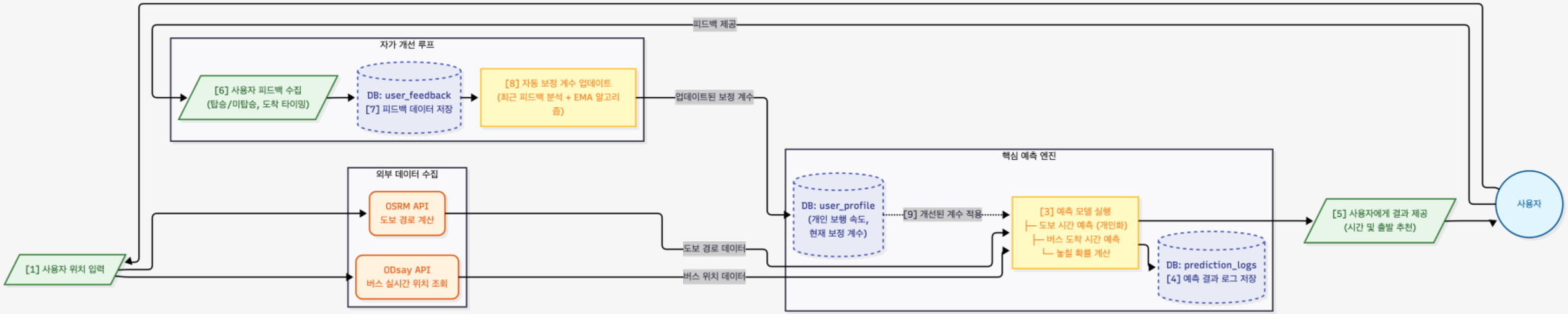
사용자 피드백

- 탑승 여부
- 타이밍 정보

모델 업데이트

- 보정 계수 업데이트
- 다음 예측 반영

04 데이터 파이프라인


피드백 제공
자가 개선 루프
[6] 사용자 피드백 수집
(탑승/미탑승, 도착 타이밍)
DB: user_feedback
[7] 피드백 데이터 저장
[8] 자동 보정 계수 업데이트
(최근 피드백 분석 + EMA 알고리즘)
업데이트된 보정 계수
외부 데이터 수집
OSRM API
도보 경로 계산
ODsay API
버스 실시간 위치 조회
[1] 사용자 위치 입력
도보 경로 데이터
버스 위치 데이터
핵심 예측 엔진
DB: user_profile
(개인 보행 속도,
현재 보정 계수)
[9] 개선된 계수 적용
[3] 예측 모델 실행
├ 도보 시간 예측 (개인화)
├ 버스 도착 시간 예측
└ 놓침 확률 계산
DB: prediction_logs
[4] 예측 결과 로그 저장
[5] 사용자에게 결과 제공
(시간 및 출발 추천)
사용자

04 데이터 수집

사용자 프로필 데이터

user_profile

개인 맞춤형 예측을 위한 사용자 특성 데이터

예측 로그 수집

prediction_logs

사용자가 버스 도착 예측을 요청할 때마다 수집

사용자 피드백 수집

user_feedback

예측 후 실제 결과를 수집하여 모델 개선에 활용

실시간 버스 위치 데이터

vehicle_positions

ODsay API를 통해 실시간 버스 위치 정보를 수집



05 데이터 처리 및 분석

보행 속도 분석

$$\text{예측 기록별 보행 속도} = \frac{\text{거리}}{\text{도착 시각 - 예측 생성 시각}}$$

Exponential Moving Average (EMA)

* α = 0.3 (최근 20 기록에 30% 가중치 부여)

$$\text{새로운 보행 속도} = \alpha \times \text{최근 보행 속도의 평균} + (1 - \alpha) \times \text{기존 사용자 보행 속도}$$

업데이트

- 실제 보행 속도 데이터로 저장된 사용자별 보행 속도를 지속적으로 업데이트
- EMA로 이상치(유난히 빠르게/느리게 걸은 날)가 큰 영향을 끼치지 않도록 방지

시간 보정 계수 계산

$$\text{실제 소요 시간} = \frac{\text{거리}}{\text{저장된 사용자 속도}} \times \text{시간 보정 계수}$$

Exponential Moving Average (EMA)

* α = 0.3 (최근 20 기록에 30% 가중치 부여)

$$\text{새로운 시간 보정 계수} = \alpha \times \text{최근 피드백의 시간 보정 계수 평균} + (1 - \alpha) \times \text{기존 설정된 시간 보정 계수}$$

업데이트

05 데이터 처리 및 분석

통계 분석 자료

최근 7일 통계

전체 예측 횟수
total_predictions

전체 피드백 수
total_feedbacks

놓침율
miss_rate
= (탑승 실패 수) / (전체 피드백 수)

온타임 탑승률
on_time_rate
= (1분 이내 대기 수) / (전체 피드백 수)

피드백 선택 통계
도착 타이밍 및 타이밍 정도 분포

06 예측 모델

입력 데이터

- 출발지/도착지 좌표
- 사용자 보행 속도
- 시간 보정 계수
- 뛰기 모드 여부

도보 경로 계산

도보 경로 API
(OSRM API)
↓
도보 경로 및
경로 거리 추출

도보 시간 계산

보정된 도보 시간
↓
보행 속도 분석
+
시간 보정 계수 적용

확률별 소요 시간 계산

도착 확률별
(50% / 99%)
소요 시간 계산

06 예측 모델

전체 불확실성

total_std_dev =
√(walk_std_dev² +
bus_std_dev²)

가정

정류소에
더 빠르게/느리게
도착한 시간차
~ 정규분포

도보 시간 변동성

walk_std_dev
= walk_eta
× walk_variance

- slow: 30% 변동성
- medium: 15% 변동성
- fast: 20% 변동성

버스 도착 변동성

bus_std_dev
= vehicle_eta
× bus_variance

- 1km 이내: 40% 변동성
- 1-3km: 30% 변동성
- 3km 이상: 20% 변동성

놓칠 확률 계산

정규분포
누적분포함수로
근사하여 계산

- 50% 분위수 = 0.6745
- 95% 분위수 = 1.96
- 99% 분위수 = 2.576

출발 시점 추천

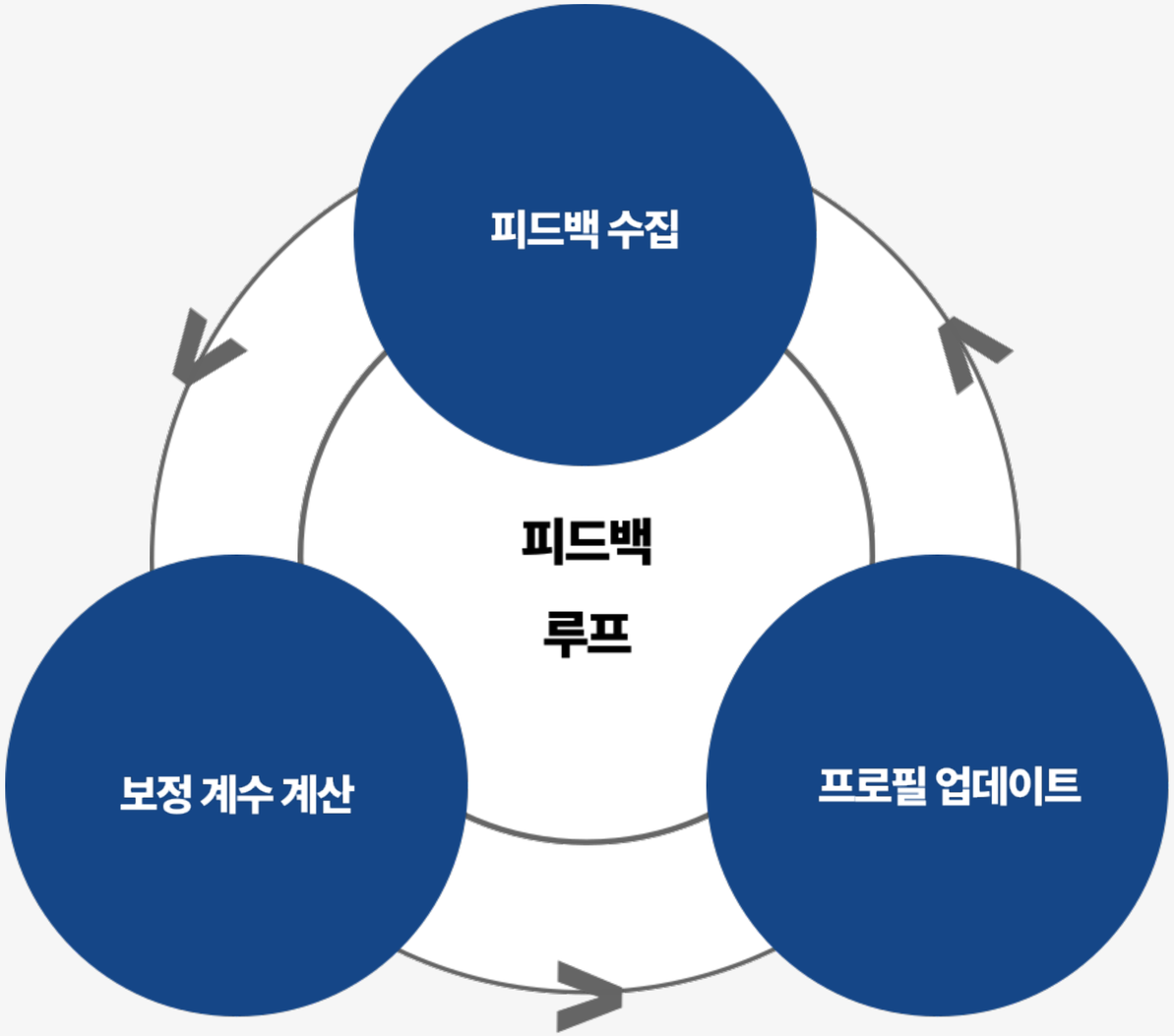
계산된 변동성 기반
버스를 놓칠 확률별
출발 시점을 추천

확률 추천
= max(0, vehicle_eta -
walk_eta - buffer)

06 예측 모델

모델 업데이트

사용자의 피드백을 반영해
도보 시간 예측에 활용되는
사용자별 시간 보정 계수를
지속해서 업데이트



07 데이터 엔지니어링 관점의 핵심 요소들

데이터 수집 자동화
모든 예측과 피드백이 자동으로 데이터베이스에 저장

확률 기반 의사결정
정규분포 기반 확률 계산으로 신뢰도 높은 예측 제공

실시간 데이터 처리
버스 위치 정보를 실시간으로 조회하여 예측에 반영

개인 맞춤화
사용자별 보행 속도 및 보정 계수를 통한 정확한 예측

지속적 학습
피드백 기반 자동 모델 개선으로 예측 정확도 향상

통계 분석
놓침율, 온타임 탑승률 등 핵심 지표 추적

08 사용자 후기

제 걸음걸이가 일반 성인 평균보다 조금 느린 편이라
처음에는 오차가 있었는데, 앱을 계속 쓰다 보니
예측 시간이 점점 맞아떨어지기 시작했습니다.
데이터가 쌓일수록 나에게 최적화된다는 느낌을 받았습니다.

내 보행 속도를 고려해서 소요 시간을
알려준다는 점에서 바쁜 현대인에게
도움이 되는 서비스 같습니다.

총 15명의 사용자

개인 맞춤형으로 평상시와 급할 때(뛰어야 할때)를
구분해서 시간을 제공받을 수 있어서 좋은 것 같습니다.

확률별 출발 추천 시간을 알려주니까
내 상황에 맞게 이번 버스를 탈지,
보내고 다음 버스를 탈지
유동적으로 선택할 수 있는 선택지가 생겨 좋았습니다.

08 사용자 후기

혹시 버스를 놓치게 되었을 때 대안을 제시해주면 좋을 것 같습니다!!

"뛰기" 모드로 계산할 때 **달리면서 지쳐,**
속도가 늦어질 수 있다는걸 고려한다면 좋을 것 같습니다.

날씨 데이터와 연동해서 '우천 시 보정된 예상 시간'을 알려준다면
데이터 활용도가 더 높아질 것 같습니다.

뛰는 시간을 계산해 주는 기능은 정말 신선했습니다.
다만, 달리는 도중에 횡단보도 신호에 걸리는 시간이나,
계단을 오르내리는 시간 같은 **'변수'들도 반영**된다면 완벽할 것
같습니다. 지금 상태로도 충분히 기존 앱보다 유용합니다!

09 향후 개선 방향

01

머신러닝 모델 도입

뛰기 모드 개선을 위해 경사도 고려
체력으로 인해 속도가 느려지는 것 고려

02

A/B 테스팅

EMA 알고리즘의 α값, 데이터 처리 알고리즘 수정 등
다양한 예측 알고리즘 성능 비교

03

배치 처리

대량 데이터 분석을 위한
배치 처리 파이프라인 구축

04

실시간 스트리밍

버스 위치 데이터 실시간 스트리밍 처리

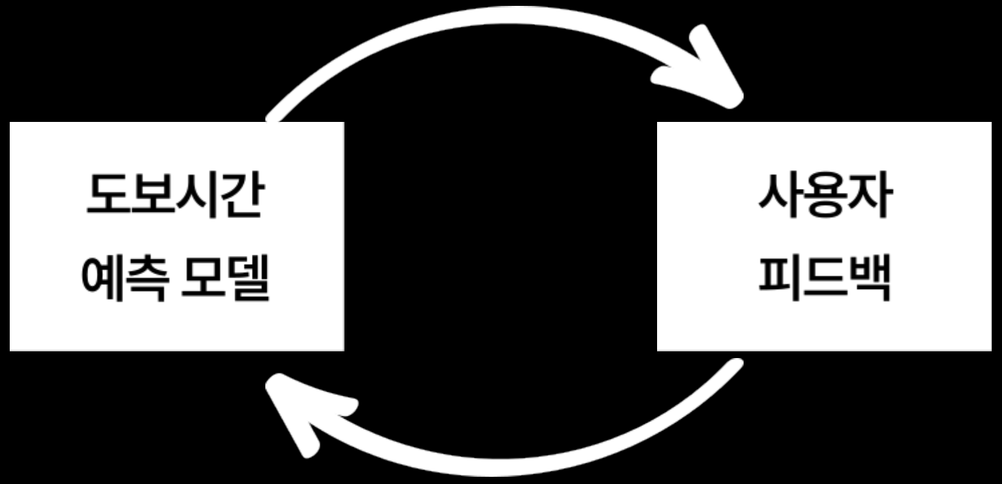
감사합니다.

Appendix

실시간 버스 정류소
도보 시간 예측



사용자 피드백 기반
지속적 학습 시스템



데이터 수집

예측 로그 수집

(`prediction_logs`)

사용자가 버스 도착 예측을 요청할 때마다 수집

- 위치 데이터
  - 출발지 좌표
- 정류장 정보
  - 정류장 ID
  - 노선 ID
- 예측 데이터
  - 도보 예상 시간
  - 버스 도착 예상 시간
  - 추천 출발 시간
- 확률 데이터
  - 50% 놓칠 확률
  - 95% 놓칠 확률
- 메타데이터
  - 모델 버전
  - 생성 시각

데이터 수집

사용자 피드백 수집

(`user_feedback`)

예측 후 실제 결과를 수집하여 모델 개선에 활용

- 탑승 결과
  - 탑승 성공 여부
- 타이밍 정보
  - 도착 타이밍
  - 타이밍 정도
- 대기 시간
  - 추가 대기 시간
- 경로 정보
  - 도보 경로 거리
  - 예상 도보 시간
- 참조 데이터
  - 예측 로그 ID

데이터 수집

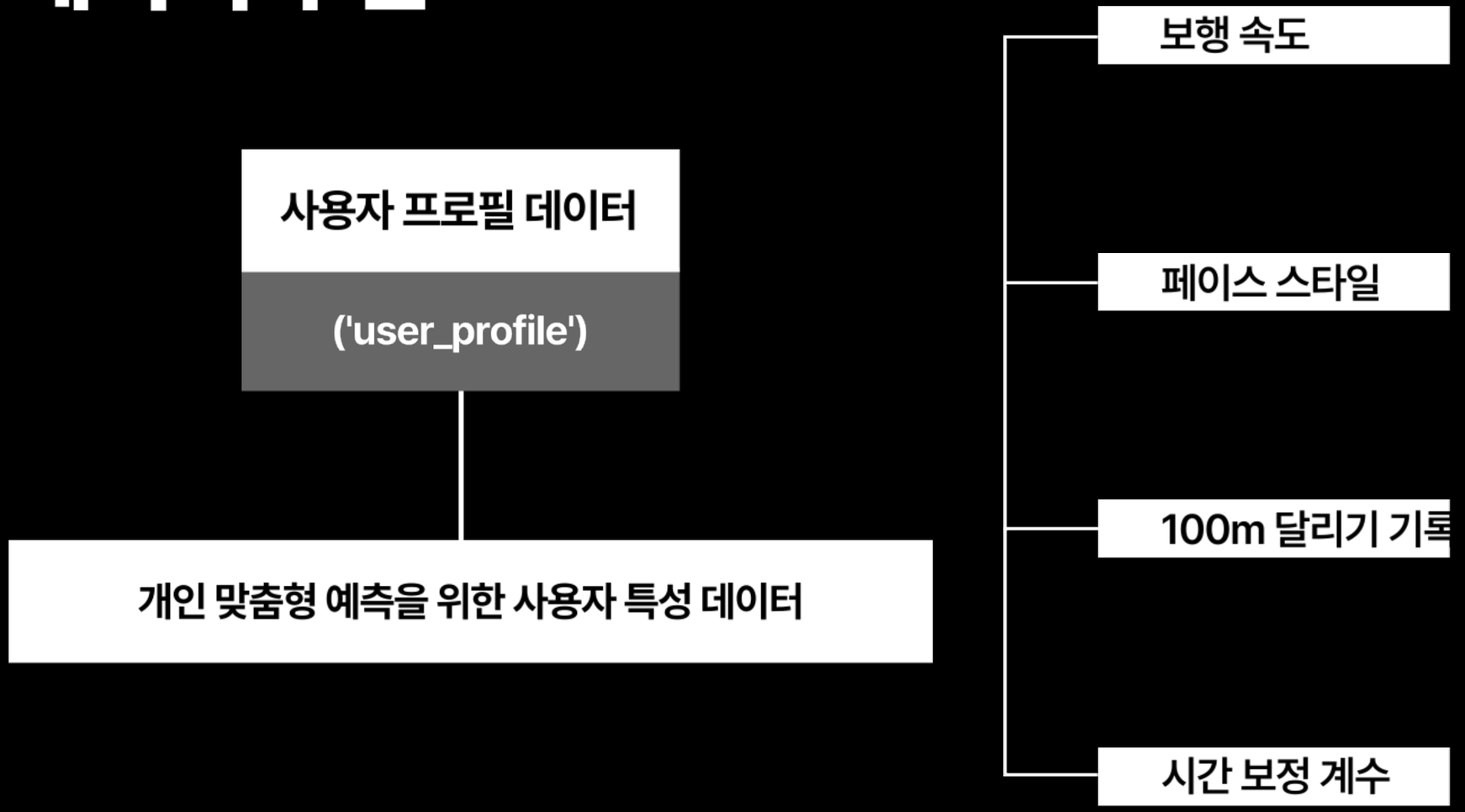
실시간 버스 위치 데이터

('vehicle_positions')

ODsay API를 통해 실시간 버스 위치 정보를 수집

- 차량 ID
- 노선 정보
- GPS 좌표
- 다음 정류장 정보
- 혼잡도 정보

데이터 수집


사용자 프로필 데이터
('user_profile')
개인 맞춤형 예측을 위한 사용자 특성 데이터
보행 속도
페이스 스타일
100m 달리기 기록
시간 보정 계수