

# 당뇨 위험도 예측 앱

Flutter + FastAPI 기반의 당뇨 위험도 예측 모바일 앱입니다.  
사용자가 간단한 건강 정보를 입력하면 머신러닝 모델이 당뇨 위험 확률을 분석하고, 필요 시 주변 병원 검색과 길찾기까지 연결해 줍니다.

## 주요 기능

### 당뇨 위험도 예측

- **간편 예측:** 나이, 키/몸무게(BMI), 임신횟수, 혈당을 라디오 버튼 구간으로 선택
- **상세 예측:** 각 항목을 직접 수치로 입력
- 혈당 수치는 선택 사항이며, 입력 여부에 따라 서로 다른 모델이 적용됨
- 예측 결과를 확률, 판정, 차트 이미지로 제공

### 병원 검색 및 길찾기

- 주소 검색 후 좌표 기반으로 주변 병원 목록 조회 (공공데이터 API)
- 병원 카드에서 길찾기 버튼을 누르면 카카오맵, 네이버지도, 티맵, Apple Maps 등 설치된 지도 앱으로 바로 연결

### 기타

- 다크 모드 / 라이트 모드 전환
- API 서버 주소 사용자 지정 (실기기 테스트 대응)
- 주소 및 좌표 로컬 저장

## 기술 스택

구분	기술
프론트엔드	Flutter (Dart), Material Design 3
백엔드	FastAPI (Python)
ML 모델	scikit-learn (LR, SVM, Voting Ensemble)
데이터 시각화	Matplotlib (서버 사이드 차트 생성, Base64 전송)
주소 검색	카카오 주소검색 API (kpostal)
병원 조회	공공데이터 건강보험심사평가원 API
좌표 변환	Nominatim (geopy)
지도 연동	map_launcher
로컬 저장소	GetStorage

## 프로젝트 구조

```

diabetes_app/
├── lib/
│   ├── main.dart                # 앱 진입점
│   ├── config.dart             # API 키 및 기본 URL 설정
│   ├── view/
│   │   ├── main_tab_page.dart  # 메인 탭 (간편/상세 예측 전환)
│   │   ├── simple_predict_page.dart # 간편 예측 화면
│   │   ├── detail_predict_page.dart # 상세 예측 화면
│   │   ├── hospital_search_page.dart # 병원 검색 + 길찾기
│   │   └── address_search_page.dart # 주소 검색 + 좌표 변환
│   ├── models/
│   │   ├── hospital.dart       # 병원 데이터 모델
│   │   └── predict_input_profile.dart # 예측 입력 프로필 모델
│   ├── widgets/
│   │   ├── app_settings_drawer.dart # 설정 드로어 (테마, API URL)
│   │   ├── age_picker.dart      # 나이 선택 ( Cupertino 휠 )
│   │   ├── height_weight_picker.dart # 키/몸무게 입력 및 BMI 자동 계산
│   │   └── percentile_range_radio.dart # 분위 구간 라디오 버튼
│   ├── constants/
│   │   └── diabetes_predict_mapping.dart # 혈당/임신횟수 구간 매핑
│   ├── utils/
│   │   ├── app_storage.dart     # GetStorage 래퍼
│   │   ├── custom_common_util.dart # 공통 유틸 (로딩, 스낵바, 검증 등)
│   │   ├── json/custom_json_util.dart # JSON 파싱/변환 유틸
│   │   └── xml/custom_xml_util.dart # XML 파싱/변환 유틸
│   ├── navigation/
│   │   └── custom_navigation_util.dart # 커스텀 페이지 전환
│   └── theme/
│       ├── app_theme_colors.dart # 라이트/다크 테마 색상 정의
│       └── theme_provider.dart   # 테마 상태 관리
├── fastapi/
│   ├── app/
│   │   ├── main.py              # FastAPI 앱 (엔드포인트 정의)
│   │   ├── schemas.py           # Pydantic 요청/응답 스키마
│   │   ├── predictor.py         # 예측 로직 + 차트 생성
│   │   ├── model_loader.py      # 4시나리오 모델/전처리 로더
│   │   ├── geocoding.py         # 주소 → 좌표 변환
│   │   ├── model_sugar.joblib   # 런타임 호환 모델 (Scenario A)
│   │   ├── model_no_sugar.joblib # 런타임 호환 모델 (Scenario B)
│   │   ├── a_detail_sugar_model.joblib # A: 상세/수치형(혈당 포함)
│   │   ├── b_detail_no_sugar_model.joblib # B: 상세/수치형(혈당 미포함)
│   │   ├── c_simple_sugar_model.joblib # C: 심플/등급형(혈당 포함)
│   │   └── cns_simple_no_sugar_model.joblib # C-NS: 심플/등급형(혈당 미포함)
│   ├── requirements.txt
│   ├── APIGUIDE.md              # API 명세 문서
│   └── scripts/                  # 학습/검증 스크립트
├── android/                      # Android 플랫폼
├── ios/                          # iOS 플랫폼
└── pubspec.yaml                  # Flutter 의존성

```

---

## 실행 방법

### 1. FastAPI 백엔드 서버

```
cd fastapi
pip install -r requirements.txt
python -m uvicorn app.main:app --reload --host 0.0.0.0 --port 8000
```

서버 실행 후 <http://localhost:8000/docs>에서 Swagger UI로 API를 테스트할 수 있습니다.

### 2. Flutter 앱

```
flutter pub get
flutter run
```

실기에서 테스트할 경우, 앱 설정 드로어에서 API 서버 주소를 PC의 로컬 IP로 변경해야 합니다.  
서버의 [/health](#) 엔드포인트 응답에서 [suggested\\_url](#)을 확인할 수 있습니다.

---

## API 엔드포인트

Method	URL	설명
GET	<a href="#">/health</a>	서버 상태 및 모델 정보 확인
POST	<a href="#">/predict</a>	당뇨 위험도 예측 (확률, 판정, 차트)
POST	<a href="#">/geocode</a>	한글 주소 → 위도/경도 변환

요청/응답 상세는 [APIGUIDE.md](#)를 참고하세요.

---

## ML 모델

Pima Indians Diabetes Dataset(당뇨.csv)을 기반으로 학습한 4개 시나리오 모델을 사용합니다.

시나리오	모델	입력 피쳐	Test Accuracy
A (상세/혈당 포함)	LogisticRegression	임신허수, 혈당, BMI, 나이	0.7403
B (상세/혈당 미포함)	SVC	임신허수, BMI, 나이	0.6753
C (심플/혈당 포함)	Voting Ensemble (Top 3 Mix)	임신허수, 혈당, BMI, 나이	0.7273
C-NS (심플/혈당 미포함)	Voting Ensemble (Top 3 Mix)	임신허수, BMI, 나이	0.6818

API는 [입력모드\(detail/simple\)](#) + 혈당 입력 유무로 A/B/C/C-NS를 분기하며, 시나리오별 전처리(IQR clipping, scaler/imputer, 등급화)와 임계값을 적용합니다.

---

## 네이티브 권한 설정

외부 지도 앱 호출을 위해 플랫폼별 설정이 필요합니다.

**iOS** (`ios/Runner/Info.plist`) - `LSApplicationQueriesSchemes`에 `kakaomap`, `nmap`, `tmap`, `comgooglemaps`, `maps` 등록

**Android** (`android/app/src/main/AndroidManifest.xml`) - `<queries>` 블록에 `kakaomap`, `nmap`, `tmap` 인텐트 선언

---

## 개발 환경

- Flutter SDK  $\geq 3.10.8$
- Python 3.10+
- scikit-learn, FastAPI, uvicorn
- 실기기 테스트 권장 (지도 앱 연동은 시뮬레이터에서 불가)