

User Manual

Diabetes Health App



MAY 1

Diabetes Health App
Authored by: Group 10

Diabetes Health App

Introduction

Diabetes is a worldwide health problem that have affected 400 million people worldwide. There are two major types of diabetes:

- Type 1- often diagnosed in childhood.
- Type 2- largely caused by an unhealthy lifestyle and diet.

For Singapore, from a study done in 2015, it had the second highest proportion of diabetes among developed nations. The number of diabetic cases were projected to further increase by another 200K to 600K by 2030. In terms of long-term complications, we had the highest rate of lower limb amputation in the world with about 1200 diabetes undergoing amputation every year.

While there is no cure for diabetes, it can be managed and prevented for type 2 diabetes. By leveraging the power of machine learning and Artificial Intelligence ('AI'), this project intend to identify customer who is at risk of diabetes, look out for evidence of long-term complications such as blindness and leg amputation and manage diabetes through one's lifestyle (i.e., exercise and dietary).

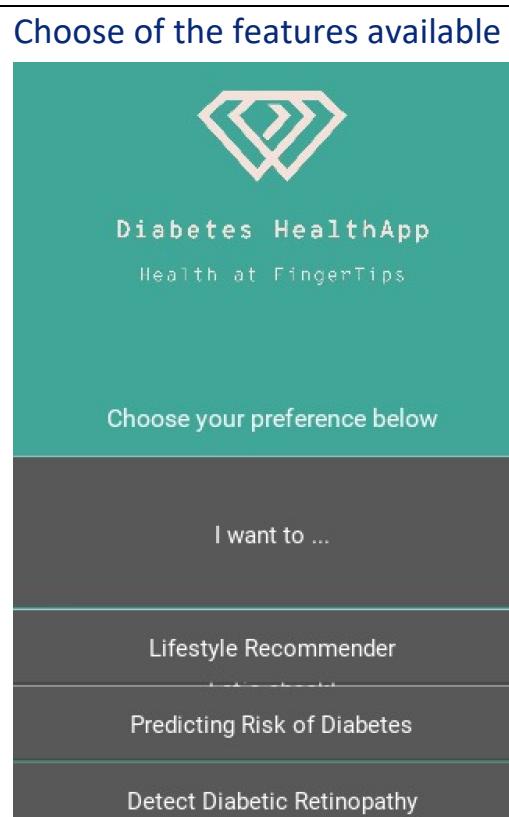
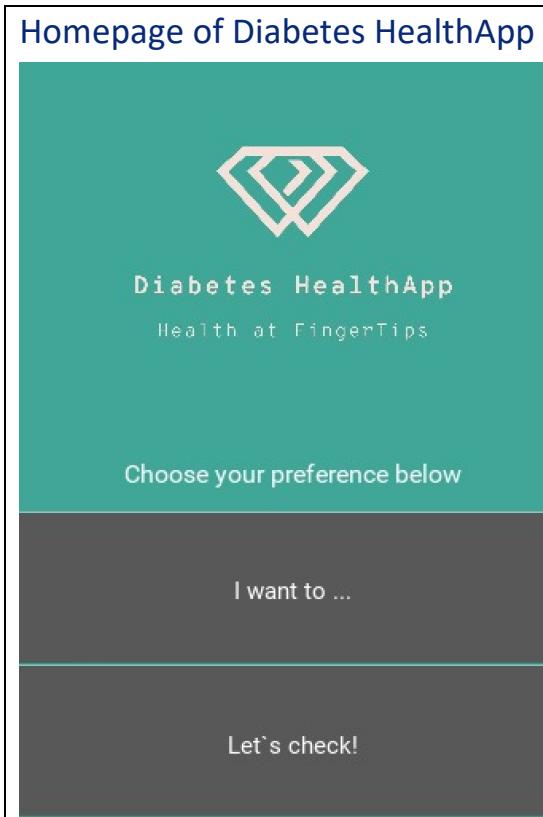
The project team has adopted an Agile philosophy in approaching the problem statement – prioritizing the attainment of a first-generation release of the product as swiftly as possible, focusing on the core functionality over pursuing a more “perfect” product. In this regard, the first release should be taken as a “minimum viable product” (MVP) that will be improved upon in successive releases with more use experience. The product has the potential to cover the full diabetes life cycle and build new verticals to detect health problems (i.e., Heart Diseases, Diabetes, Thyroid disorder, Cancer, and Multiple sclerosis) though our AI-enabled eye scanner.

User Interface Guide

App guide

There are 3 main features in the app.

1. Predicting Risk of Diabetes
2. Lifestyle Recommender
3. DR Detection



Get some info from users

Life Style

Age (YEARS):

Exercise frequency
(days per week):

28

2

Height (CM):

Exercise duration
(hours per day):

176

1

Weight (KG):

Breakfast:

70

Pancakes, plain; Pancake syrup, NFS

Activity:

Lunch:

dancing

Chicken with mole sauce

Category:

Dinner:

bench step class, general

Catfish, coated, fried, made with oil

Outcome

Go to Home page

Return some advice!!

Diabetic Health Report



~~~Some advices for you~~~

"breakfastMealAdvice": "Please get healthier breakfast"

"lunchMealAdvice": "Please get healthier lunch"

"dinnerMealAdvice": "Please get healthier dinner"

"physicalExerciseAdvice": "Please increase your exercise activity for a healthier you"

For more detail please watch our video!!

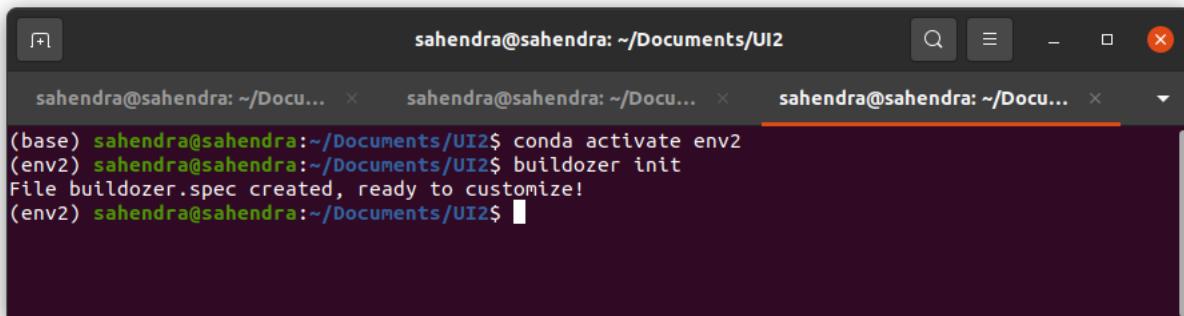
# Installation Guide

## User Interface (App)

The app is build with Android Studio and python

Following is step by step guides on how to deploy the code:

1. Install buildozer (skip if this had been installed in your system)
  - Pip install buildozer
2. Navigate to project directory (it located in project drive SystemCode/HealthApp
3. Execute following command
  - Buildozer init

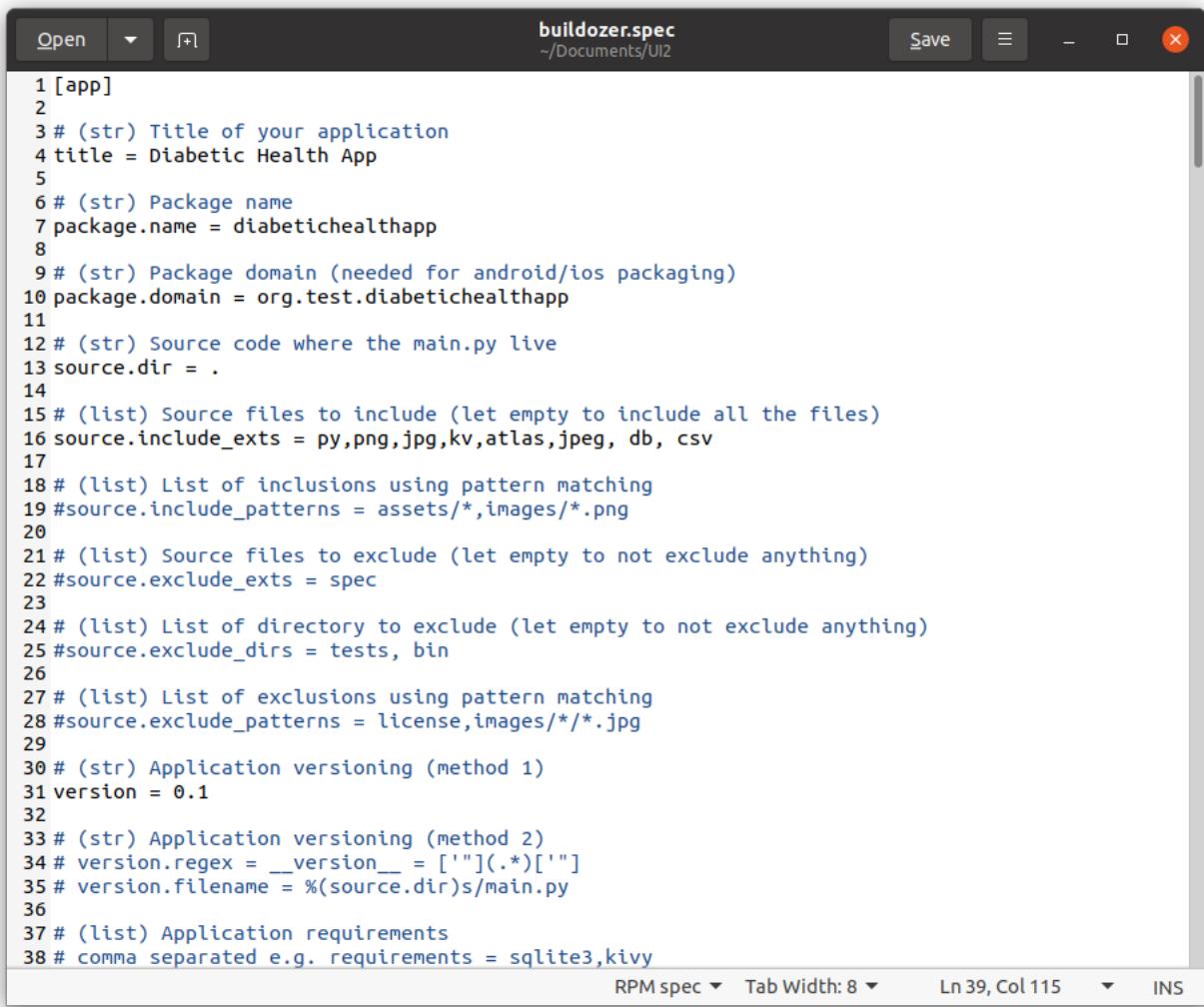


A screenshot of a terminal window titled "sahendra@sahendra: ~/Documents/UI2". It shows three tabs open. The active tab displays the command "buildozer init" being run, with the output: "(base) sahendra@sahendra:~/Documents/UI2\$ conda activate env2 (env2) sahendra@sahendra:~/Documents/UI2\$ buildozer init File buildozer.spec created, ready to customize! (env2) sahendra@sahendra:~/Documents/UI2\$". The terminal has a dark background with light-colored text.

4. Step 3 will generate buildozer.spec file
5. Open and edit the buildozer.spec file

| Properties          | Values                                                                                             |
|---------------------|----------------------------------------------------------------------------------------------------|
| title               | Diabetes Health App                                                                                |
| package.name        | diabeteshealthapp                                                                                  |
| package.domain      | org.test.diabeteshealthapp                                                                         |
| source.include_exts | py,png,jpg,kv,atlas,jpeg, db, csv                                                                  |
| requirements        | python3,kivy, kivymd, Pillow, sqlite3, pandas, numpy, requests, urllib3, charset_normalizer, idna, |

|                     |                                                               |
|---------------------|---------------------------------------------------------------|
| android.permissions | INTERNET,<br>READ_EXTERNAL_STORAGE,<br>WRITE_EXTERNAL_STORAGE |
|---------------------|---------------------------------------------------------------|



```

1 [app]
2
3 # (str) Title of your application
4 title = Diabetic Health App
5
6 # (str) Package name
7 package.name = diabetichealthapp
8
9 # (str) Package domain (needed for android/ios packaging)
10 package.domain = org.test.diabetichealthapp
11
12 # (str) Source code where the main.py live
13 source.dir = .
14
15 # (list) Source files to include (let empty to include all the files)
16 source.include_exts = py,png,jpg,kv,atlas,jpeg, db, csv
17
18 # (list) List of inclusions using pattern matching
19 #source.include_patterns = assets/*,images/*.png
20
21 # (list) Source files to exclude (let empty to not exclude anything)
22 #source.exclude_exts = spec
23
24 # (list) List of directory to exclude (let empty to not exclude anything)
25 #source.exclude_dirs = tests, bin
26
27 # (list) List of exclusions using pattern matching
28 #source.exclude_patterns = license,images/*/*.jpg
29
30 # (str) Application versioning (method 1)
31 version = 0.1
32
33 # (str) Application versioning (method 2)
34 # version.regex = __version__ = ['"](.*)['"]
35 # version.filename = %(source.dir)s/main.py
36
37 # (list) Application requirements
38 # comma separated e.g. requirements = sqlite3,kivy

```

RPM spec ▾ Tab Width: 8 ▾ Ln 39, Col 115 ▾ INS

## 6. Execute command as below

- Buildozer -v android debug

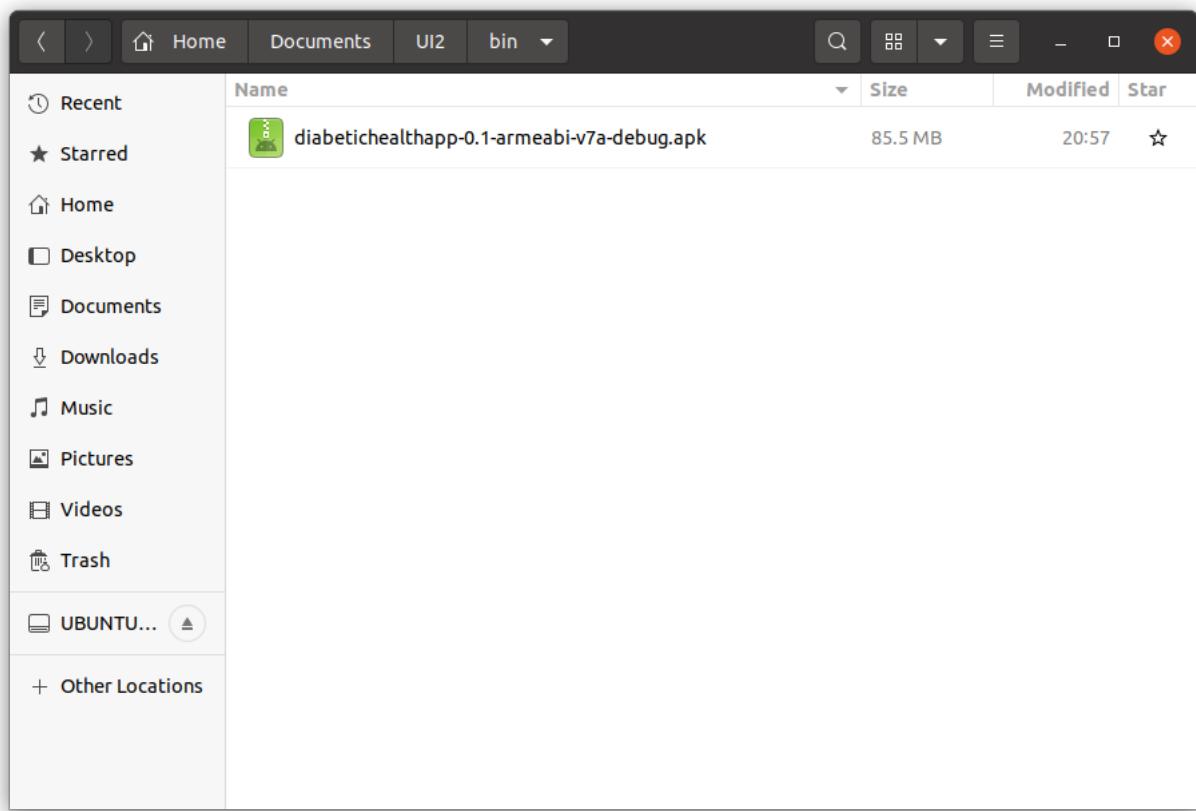
```
sahendra@sahendra: ~/Documents/UI2
(base) sahendra@sahendra:~/Documents/UI2$ conda activate env2
(env2) sahendra@sahendra:~/Documents/UI2$ buildozer int
# Check configuration tokens
ERROR: You already have a buildozer.spec file.
(env2) sahendra@sahendra:~/Documents/UI2$ buildozer -v android debug
```

## 7. Observed the process until it finished and the log indicated Android Packaging done!

```
sahendra@sahendra: ~/Documents/UI2
[DEBUG]:      /home/sahendra/.buildozer/android/platform/android-ndk-r19c/toolchains/arm-linux-androideabi-4.9/prebuilt/linux-x86_64/bin/arm-linux-androideabi-strip:/home/sahendra/Documents/UI2/.buildozer/android/platform/build-armeabi-v7a/dists/diabetichealthapp/build/intermediates/merged_native_libs/debug/out/lib/armeabi-v7a/gdb.setup: File format not recognized
[DEBUG]:      Unable to strip library /home/sahendra/Documents/UI2/.buildozer/android/platform/build-armeabi-v7a/dists/diabetichealthapp/build/intermediates/merged_native_libs/debug/out/lib/armeabi-v7a/gdb.setup due to error 1 returned from /home/sahendra/.buildozer/android/platform/android-ndk-r19c/toolchains/arm-linux-androideabi-4.9/prebuilt/linux-x86_64/bin/arm-linux-androideabi-strip, packaging it as is.
[DEBUG]:
[DEBUG]:      Deprecated Gradle features were used in this build, making it incompatible with Gradle 7.0.
[DEBUG]:      Use '--warning-mode all' to show the individual deprecation warnings.
[DEBUG]:      See https://docs.gradle.org/6.4.1/userguide/command_line_interface.html#sec:command_line_warnings
[DEBUG]:      > :packageDebug
[DEBUG]:      BUILD SUCCESSFUL in 11s
[DEBUG]:      25 actionable tasks: 25 executed

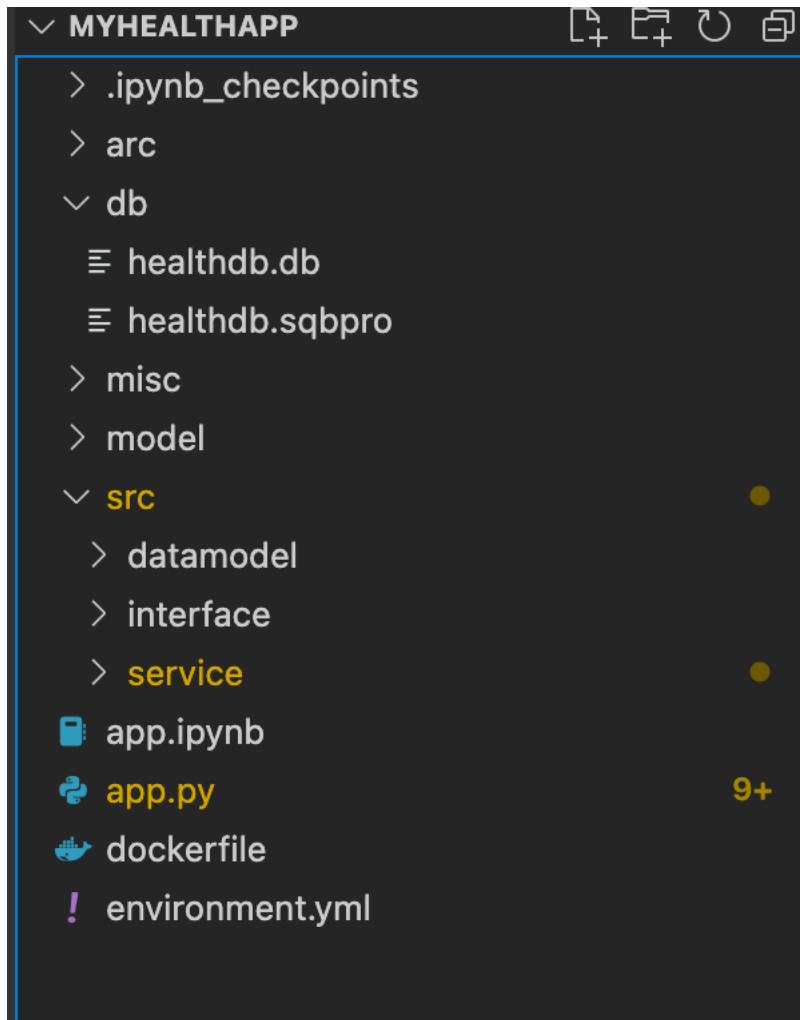
[INFO]:      <- directory context /home/sahendra/Documents/UI2/.buildozer/android/platform/python-for-android
[INFO]:      Of the existing distributions, the following meet the given requirements:
[INFO]:          diabetichealthapp: min API 21, includes recipes (freetype, hostpython3, jpeg, libbz2, libffi, liblzma, openssl, png, sdl2_image, sdl2_mixer, sdl2_ttf, sqlite3, python3, sdl2, pytz, setuptools, cython, pillow, six, numpy, pygments, android, kivy, pandas, , python-dateutil, requests, charset_normalizer, kivynd, urllib3, certifi, idna)
[INFO]:          diabetichealthapp has compatible recipes, using this one
[INFO]:      # Copying android package to current directory
[INFO]:      # Android package filename not found in build output. Guessing...
[INFO]:      # Found android package file: /home/sahendra/Documents/UI2/.buildozer/android/platform/build-armeabi-v7a/dists/diabetichealthapp/build/outputs/apk/debug/diabetichealthapp-debug.apk
[INFO]:      # Add version number to android package
[INFO]:      # Android package renamed to diabetichealthapp-debug-0.1-.apk
[DEBUG]:      --> running cp /home/sahendra/Documents/UI2/.buildozer/android/platform/build-armeabi-v7a/dists/diabetichealthapp/build/outputs/apk/debug/diabetichealthapp-debug.apk diabetichealthapp-debug-0.1-.apk
WARNING: Received a --sdk argument, but this argument is deprecated and does nothing.
No setup.py/pyproject.toml used, copying full private data into .apk.
Applying Java source code patches...
Applying patch: src/patches/SDLActivity.java.patch
# Android packaging done!
# APK diabetichealthapp-0.1-armeabi-v7a-debug.apk available in the bin directory
(env2) sahendra@sahendra:~/Documents/UI2$
```

## 8. APK file will be created in Bin Folder



## Back end services

### Project Folder



### Application Components

- Services (python based)
  - Python 3.7
  - Flask and Flask-restful
  - Pandas
  - Numpy
  - Psycopg2
  - Scikit-learn
  - Tensorflow
  - Keras
  - Pillow
  - Opencv-python-headless

- 
- Database
    - Sqlite3 (.db file)
  - Model
    - Pickle (.pkl file)
    - Hierarchical Data Format (HDF) (.h5)

## Docker Installation

In this project for backend services we deploy in docker and we use EC2 linux AWS for simplicity.

The step to get this services ready:

1. Provision a linux EC2 t2.xlarge (storage 100 GB)
2. Install docker
  - a. sudo amazon-linux-extras install docker
  - b. sudo service docker start
  - c. sudo systemctl enable docker
  - d. sudo usermod -a -G docker ec2-user

## Docker deployment

Component that important to deploy a python services in python (conda)

1. dockerfile (prepare a docker image to host the services)
2. environment.yml (to list all dependencies)

Command to run to build a docker image and to start docker services

1. Navigate to Project Folder MyHealthApp
2. Run command: docker build -t healthapp .
3. Run command: docker run -d -p 5001:5001 healthapp

## Command to check docker logs

1. Run command: docker logs -f healthapp

---

### **Command to stop docker container**

1. Find the docker id: docker container ls -aq
2. Run the stop command: docker container stop <id returned on no 1 command>

### **Command to remove docker image**

1. Find the docker id: docker image ls
2. Run the stop command: docker image rm <id returned on no 1 command>

## **Data Preparation Guide**

### Predicting Diabetes Risk

There are multiple steps being performed to provide the Naïve Bayes model, all the scripts are provided in Miscellaneous/DataPreparation/diabeticRisk

1. Data being merged and process to create a balance data. For detail please refer to data-processing.ipynb
2. Once data is merged, all the features are assessed and determine the correlation. Refer to features\_diabetes\_correlation.ipynb for process and result
3. Once features determined, the model is build and deployed with pickle. Refer to dbtmodel.ipynb.
4. Before the model is deployed, the testing of pickle file is being performed. Refer to dbtmodel-predict.ipynb

### Lifestyle Recommender

There are multiple steps being performed to provide the decision tree and KNN models to predict BMI as part of the lifestyle recommender, the script provided in Miscellaneous/DataPreparation/lifestyle. Below are the main steps (refer to DataPrep\_LS.ipynb)

- 
1. Data being merged and process as it was from 2 different data.
  2. Clean up the empty data and do data aggregation.
  3. Once data is merged, all the features are assessed and determine the correlation. Remove unnecessary feature.
  4. Once features determined, the model is build and deployed with pickle.
  5. Before the model is deployed, the testing of pickle file is being performed.

Apart from model, lifestyle also use database to store information. The data is provided in csv and uploaded to sqlite3 through import feature.

## DR Detection

The model is created by train DR photography's images of left and right eye and their label that indicate the level. For this purpose the team using tensorflow, for detail process please refer to [Miscellaneous/DataPreparation/drDetection/dataPrep\\_DR.ipynb](#).

---

# Appendix

## References

1. <https://docs.docker.com/>
2. <https://pythonspeed.com/articles/activate-conda-dockerfile/>
- 3.

## Data Source

Data is not embedded into Project Documentation but all the data being used can be obtained from below websites:

1. <https://www.kaggle.com/alextreboul/diabetes-health-indicators-dataset>
2. [www.globaldietarydatabase.org/management/microdata-surveys](http://www.globaldietarydatabase.org/management/microdata-surveys)
3. <https://www.kaggle.com/c/diabetic-retinopathy-detection>