Contents

Chapter 1

Problem statement and ideas

1.1 Problem statement

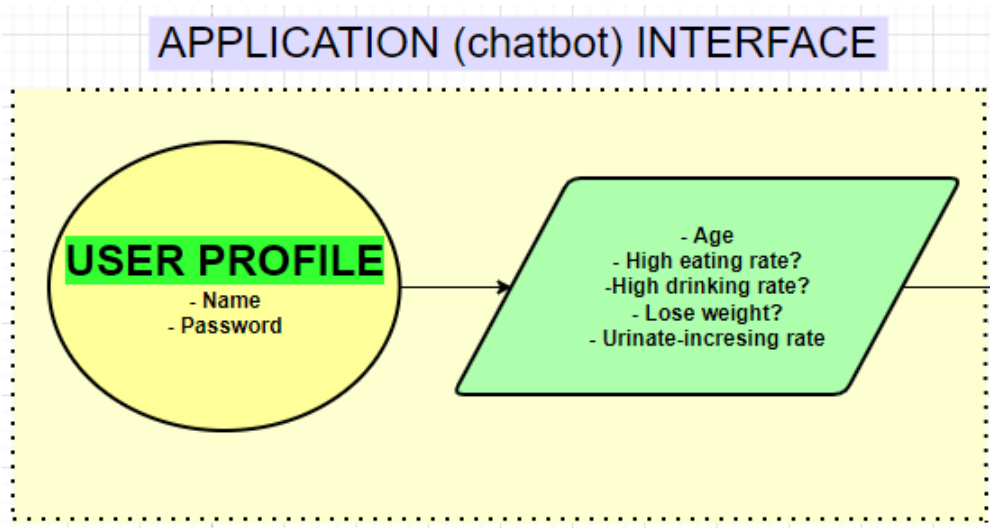
The smart doctor project development involves the use of various key components of artificial intelligence such as natural language processing, intelligent agents, machine learning, and knowledge base. Software development model Rapid Application Development (RAD) is used to design and develop a smart doctor application, with an AI-based chat-bot system to communicate with patients. The RAD model used in the system is illustrated as the following figure. The system requirements phase involves identifying the 3 main features required to incorporate into the application. The system must first understand what disease the patient has. Second, the system must inform the appropriate drug or treatment to the patient based on the diagnostic limit from various questions through the soft-bot. finally the system has to set an Alarm for the patient to take the medicine on time.

1.2 Ideas

We intend to build a project, named **KinAssist** (Kin in Kinship: family relationship, meaning our app is dedicated to assist patients with best quality, as if we are family to each other)

The system can be divided into 4 phases:

1.2.1 User Interface

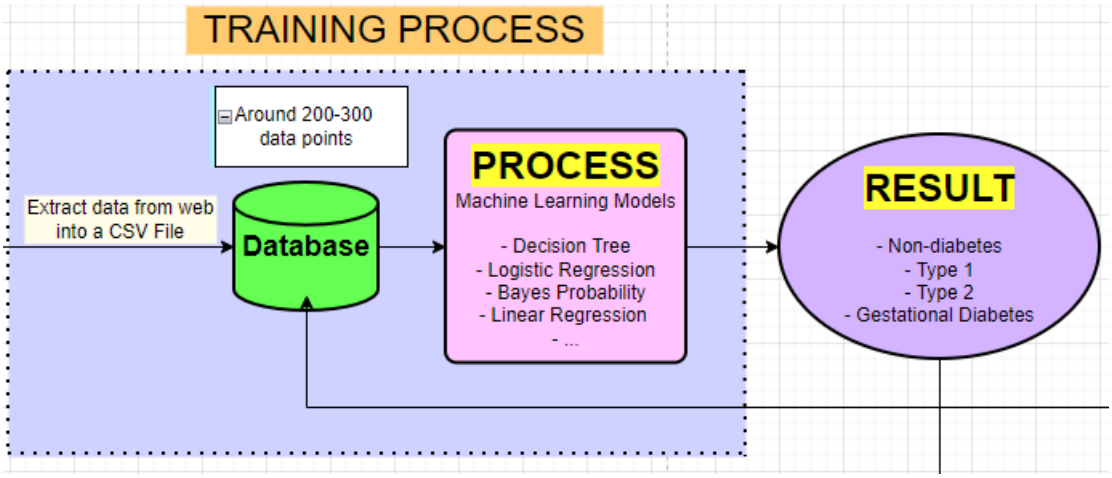


There will be an interface where patient can communicate with the chatbot (here we may use a Web interface to prototype the function of the chatbot). Basically, we are going to build a network where patient can efficiently state their current status as well as getting advices from doctors

The fact that patient needs to get to hospital or places where doctors work is sometime really hard and inconvenient. Therefore, this system will ”pretend” to be a human doctor and predict what kind of ”sickness” the patient is having (base on some symptoms) and where can they get advices, rather than letting the patients find doctors themselves. Here, for better illustration, we will focus on diabetes disease.

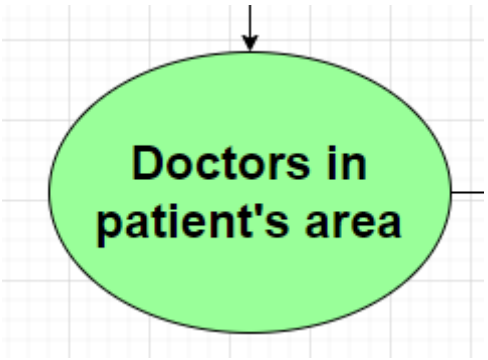
1.2.2 Project database and Machine Learning (ML) models

After having the data from patients, the next step is to predict the result of diabetes based on the given input. There are many models that can be used for this prediction, such as: **Decision Tree, Linear Regression, Logistic Regression, Bayes probability, Tree Classifier...**. These models are used almost for binary results and classification problems, for the purpose to predict whether patient doesn't have diabetes, or type 1, type 2, or gestational type.



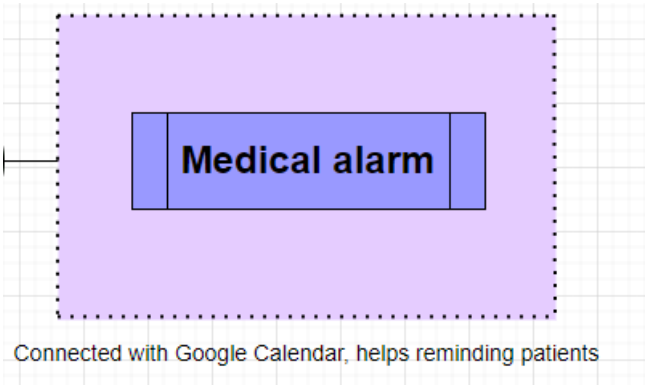
1.2.3 Finding doctors in local area

This process can be done by letting the system access to the address of doctors and locate them on the map (This is likely due to Google Map system). Then the system will receive the information of doctors that has location near to the patient (and having majors matching with problems the patient is having). This will help the patient to reduce travelling time and be more active in reaching doctors' destination.



1.2.4 Medical-taking reminder with respect to time

While many people still struggling with how can they remember when to take their medicine on time, we managed to create a small system that collabs with Google Calendar.



Chapter 2

About the project

2.1 Coding Language

Require the usage of HTML, JavaScript and Python

2.2 Process

2.2.1 Take data from user

Diabetes-types prediction

Source: SOUL_LOGIX - KinAssist

Age:
19
Sex:
Female Male
Increase in eating habit? (Yes/No)
Yes
No
Increase in drinking habit? (Yes/No)
Yes
No
Lose weight recently? (Yes/No)
Yes
No
Increase in urination? (Yes/No)
Yes
No

age	sex	eating-increasing-rate	drinking-increasing-rate	lose-weight-rate	urinate-increasing-rate
19	M	no	yes	yes	no

Submit

This is the prototype interface of the chatbot we are going to build. As we mentioned above, we mainly focus on diabetes. User may now input their age, sex, and answer some

question related to symptoms of diabetes. But, of course, only the admin of chatbot can get the data by clicking "**Download CSV**" button above the table.

2.2.2 Model Training

After taking data, we may now consider models to predict the type of diabetes (or conclude that he/she doesn't have diabetes at all). Nowadays there are numerous models which can helps us have good accuracy of predicting outcomes. The problem related to Machine Learning projects is: How good is a dataset? Is the dataset enough for training? Is the dataset reliable? Can be trust the outcome of the dataset? Is the dataset over-fitted?...

```
[21]
names = []
scores = []
for name, model in models:
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    scores.append(accuracy_score(y_test, y_pred))
    names.append(name)
tr_split = pd.DataFrame({'Name': names, 'Score': scores})
print(tr_split)
```

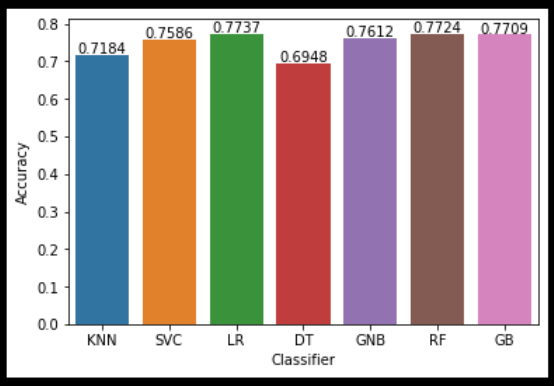
```
...   Name     Score
0  KNN  0.729282
1  SVC  0.740331
2   LR  0.790055
3   DT  0.718232
4  GNB  0.751381
5   RF  0.767956
6   GB  0.767956
```

2.2.3 Result

The output of the model will be the percentage of person having diabetes. Obviously there is a threshold for the model to categorize whether patient is non-diabetes, diabetes Type 1, Type 2 or Gestational. Also it depends on the attributed we collect from the user. The more attributes (more criteria) we collect from user, the more accuracy will the model become. Therefore, we intend to expanse the field by networking with doctors in the area and hospitals, to make the system more reliable. There still some challenges related to the dataset we need to face, frankly speaking.

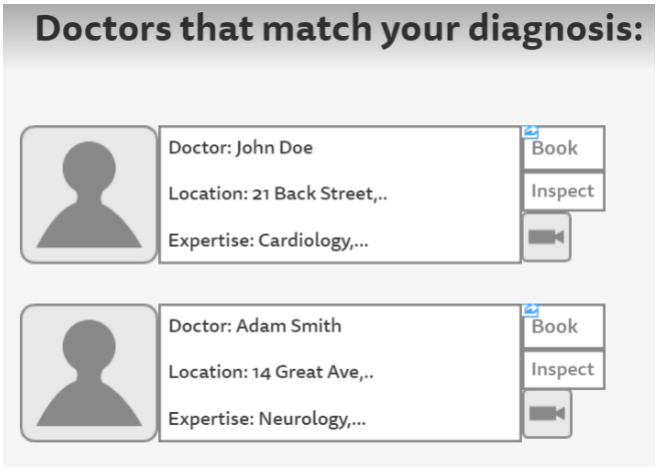
```
axis = sns.barplot(x = 'Name', y = 'Score', data = kf_cross_val)
axis.set(xlabel='Classifier', ylabel='Accuracy')
for p in axis.patches:
    height = p.get_height()
    axis.text(p.get_x() + p.get_width()/2, height + 0.005, '{:1.4f}'.format(height), ha="center")

plt.show()
```

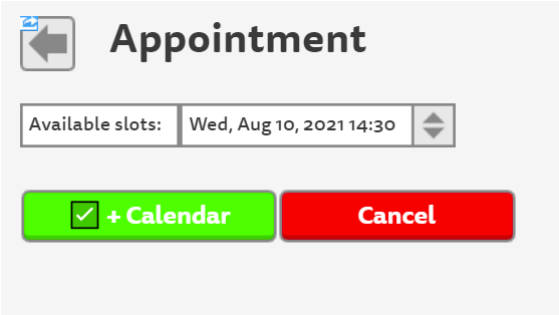
[23]



2.2.4 Doctor suggestion

Based on the results and data of the patient, the system will look up for doctors in surrounding areas that has majors related to problem the patient is having.

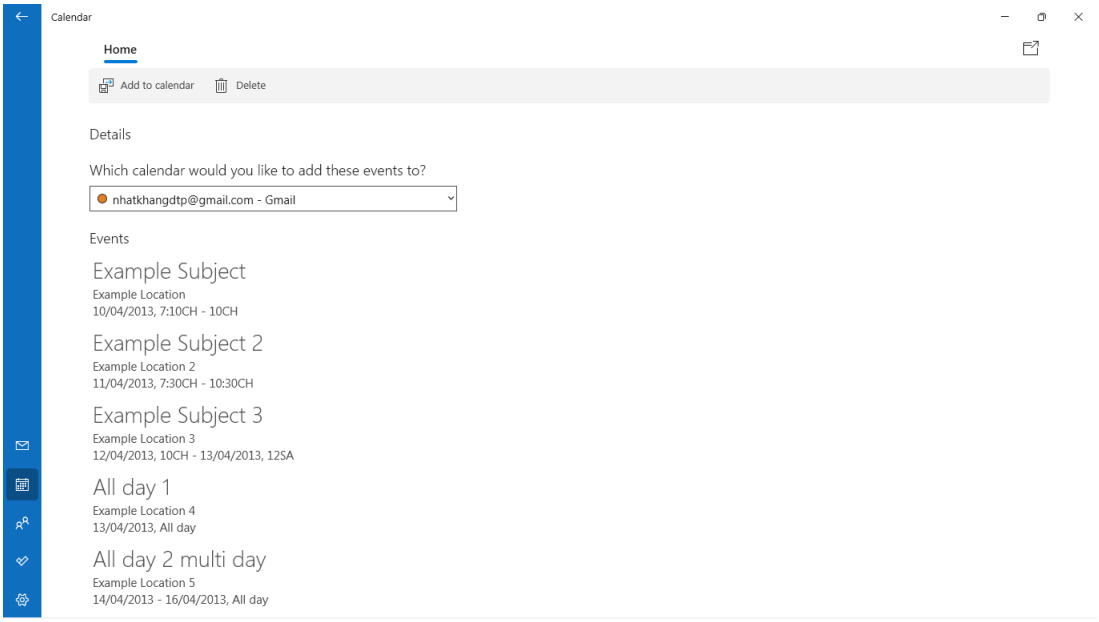




2.2.5 Medicine-taking alarm

After taking medicine from doctor, in order to make sure that patient will remember

when they are going to take medicine daily, a system is created to connect with Google Calendar and use its ability to pump up whenever it's time to take medicine. The prototype is as follow:



Chapter 3

Conclusion

3.1 Goals and Objectives

Being aware of the pressure on doctors when dealing with a large number of patients and the low efficiency of asking the same questions for every patient, our team has developed this project to collect data faster and more effectively.

Also, the system reduces travelling time and makes it easier for the patients to determine which doctors should they attend. Just use our system, patient can predict the disease rather than waiting for the doctor to get the information, securely and confidentially, and await a response. Online doctors are available at all hours of the day; if you suddenly wake up in the night feeling unwell or if you get home after a long day and feel too unwell to go out again, online medical consultation can provide you with instant answers. Moreover, making an appointment or receiving advice is never getting easier.

Many people avoid visiting the doctor because they are too embarrassed to discuss their issue; however, this can be dangerous and unhealthy if you leave a problem too long without seeking medical assistance.

When you use an online consultation service, you can talk to a trained professional from the comfort of your own home – something that can motivate people to seek help sooner. Feeling relaxed when discussing your issues is extremely important; clearly stating your symptoms and concerns is key to finding the root of the problem. When communicating with a doctor online, you can enjoy anonymity and talk freely about your medical issues without suffering from embarrassment.

3.2 Benefits for stakeholders

For stakeholders, this is a huge opportunity to expand their network of users. Once the dataset is significantly extended, better efficiency and accuracy is achieved, which is beneficial and profitable for stakeholders to collect even more data and take appropriate measures towards its customer.

3.3 Challenges and Limitations

3.3.1 Challenges

- Very big and complicated dataset, the system covers a large scope of Medical Healthcare issue
- Difficult to build as well as difficult system maintenance management.
- A great cost of money to build this system
- Data source must be reliable and not having too much error that can effect the result
- Requires an extremely good training model of Machine Learning
- Effective method to store data
- Good User Interface
- Needs to be performed well and consistently at every stage
- Always looking forward to improve model as well as making dataset more detailed

3.3.2 Limitations

- The initial data set needs to be improved so that the prediction will be more accurate. Dataset is hard to investigate.
- Need time to determine which training model is good enough
- Insufficient knowledge of Medical Healthcare
- Need time to build up a complete system