



การแจ้งเตือนเมื่อผู้สูงอายุมีอาการผิดปกติด้วย machine learning

โดย

นายพิษณุ รัตนปกรณ์ 62106935

นายศุภวิชญ์ เพชรศรี 62109665

นายศุภวิชญ์ รักยอด 62109673

อาจารย์ที่ปรึกษา

อาจารย์ ดร. ชิระวัฒน์ วัฒนพานิช

รายงานนี้เป็นส่วนหนึ่งของรายวิชา COE62-332

Artificial Intelligence

ภาคเรียนที่ 2 ปีการศึกษา 2564

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

เนื่องด้วยกลุ่มของเราได้ทำโปรเจกเกี่ยวกับการพัฒนานาฬิกาและแอปพลิเคชันเพื่อดูแลผู้สูงอายุ ดังนั้นกลุ่มพวกเราเลยสนใจที่จะใช้ machine learning ในการช่วยทำการตรวจสอบการแจ้งเตือนอาการของผู้สูงอายุว่ามีความผิดปกติไหม ก่อนที่จะส่งการแจ้งเตือนไปยังผู้ดูแลเพื่อให้ผู้ดูแลสามารถมาช่วยหรือดูแลผู้สูงอายุได้ทันทั่วทั้ง

1.2 วัตถุประสงค์ของงานวิจัย

1. เพื่อช่วยให้การแจ้งเตือนอาการผิดปกติของผู้สูงอายุมีประสิทธิภาพมากขึ้น

1.3 ขอบเขตของงานวิจัย

การใช้ Python การทำ machine learning

1.4 ประโยชน์ที่คาดว่าจะได้รับ

สามารถเพิ่มประสิทธิภาพในการดูแลผู้สูงอายุ

บทที่ 2

เอกสารและโครงการที่เกี่ยวข้อง

Machine Learning

ถ้าหากแปลตรงตัวแล้ว Machine Learning ก็คือการเรียนรู้ของเครื่องจักร ซึ่งในที่นี้เราจะต้องเข้าใจก่อนว่า Machine Learning นั้น เป็นเพียงส่วนหนึ่งของ AI ที่ช่วยให้ระบบไอทีรู้จักกับรูปแบบพื้นฐานของอัลกอริทึมและชุดข้อมูลต่าง ๆ เพื่อเป็นการเรียนรู้รูปแบบอัตโนมัติผ่านข้อมูลและประสบการณ์ด้วยตัวเอง เพื่อทำการค้นหา แยกแยะ สรุป คาดคะเน และคำนวณความน่าจะเป็น และเพื่อพัฒนากระบวนการแก้ไขปัญหาได้อย่างเหมาะสม ด้วยประสิทธิภาพการเรียนรู้ของ Machine Learning ที่มีมากขึ้นนี้ หลาย ๆ คนอาจสงสัยว่า แล้วระบบ Machine Learning มีอะไรที่สามารถนำไปปรับใช้กับการทำธุรกิจได้บ้าง

Supervised Learning เป็นการเรียนรู้ข้อมูลต่าง ๆ โดยมีผู้สอน อาศัยข้อมูลในการฝึกฝน เพื่อช่วยให้ตัวเทคโนโลยีสามารถเรียนรู้ผล และคาดคะเนผลลัพธ์ต่าง ๆ ได้อย่างแม่นยำมากยิ่งขึ้น โดยการเรียนรู้ในรูปแบบนี้ มักถูกนำมาใช้งานในเชิงธุรกิจทั้งการคำนวณราคาบ้าน การคาดคะเนค่าเงิน หรือแม้แต่การวิเคราะห์ผลการแข่งขันต่าง ๆ เป็นต้น

Unsupervised Learning เป็นการเรียนรู้ที่ไม่มีผู้สอน โดยที่เครื่องจักรนั้น ๆ จะทำการเรียนรู้สิ่งต่าง ๆ ด้วยตัวเอง โดยไม่ต้องตั้งค่าเป้าหมายของแต่ละข้อมูล ระบบสามารถนำไปวิเคราะห์และสร้างแบบแผนจากข้อมูลที่ได้รับเข้าไป มักนำไปใช้ในการแนะนำผลิตภัณฑ์ต่าง ๆ หรือคัดเลือกรายการหรือผลิตภัณฑ์ เป็นต้น

Reinforcement Learning หรือการเรียนรู้จากการลองผิดลองถูกจากสถานการณ์ในอดีตหรือระบบจำลอง เพื่อพัฒนาระบบการตัดสินใจให้ดียิ่งขึ้น เช่น การพัฒนาระบบผู้เล่นอัตโนมัติให้ชนะผู้เล่นระดับโลก ระบบการจัดการข้อมูลเพื่อนำเสนอให้ตัดสินใจเลือกอัตราส่วนของสินทรัพย์ต่าง ๆ เป็นต้น

ภาษา Python

Python คือหนึ่งในภาษาโปรแกรมระดับสูงที่ใช้กันอย่างแพร่หลาย ถูกออกแบบเพื่อให้มีโครงสร้างและ ไวยากรณ์ของภาษาที่ไม่ซับซ้อน เข้าใจง่าย มีการใช้พัฒนาแอปพลิเคชัน เว็บไซต์ รวมถึงแอปบนมือถือหรือ อุปกรณ์เคลื่อนที่ด้วย หน้าที่ของ Python ก็คือการทำงานแปลชุดคำสั่งทีละบรรทัดเพื่อป้อนเข้าสู่หน่วยประมวลผล ให้คอมพิวเตอร์ทำงานตามที่เรต้องการ หรือเรียกว่าการทำงานแบบ Interpreter นั่นเอง ด้วยภาษาที่ง่ายในการเขียน “Python” จึงมีความเหมาะสมสำหรับผู้เริ่มต้นเขียนโปรแกรมไปจนถึงนักพัฒนาในองค์กรบริษัทใหญ่ อย่างเช่น Netflix, Spotify, Google, Amazon, และ Facebook เป็นต้น

NumPy

NumPy เป็น Library พื้นฐานที่ใช้คำนวณทางคณิตศาสตร์ด้วยภาษา Python สามารถคำนวณ หรือ ดำเนินการทางตรรกะใน Array หลายมิติ หรือ Matrix ได้อย่างรวดเร็ว เพราะ Library เขียนด้วยภาษา C ที่ Compile ไว้แล้ว

Pandas

Pandas เป็น software library ที่ถูกเขียนด้วยภาษา Python ใช้ในการจัดการและวิเคราะห์ข้อมูล หรือ data โดยมี data structures และ operations ต่างๆ ที่ใช้ในการจัดการกับตารางตัวเลขและ time series

scikit-learn

scikit-learn เป็นไลบรารีฟรีในภาษาไพธอนสำหรับการพัฒนาโปรแกรมโดยใช้การเรียนรู้ของเครื่อง[3] จุดเด่นคือฟังก์ชันในการแบ่งประเภทข้อมูล การแบ่งกลุ่มข้อมูล การวิเคราะห์การถดถอยหลายอย่างไม่ว่าจะเป็น ซัพพอร์ตเวกเตอร์แมชชีน การเรียนรู้ต้นไม้ตัดสินใจ และการแบ่งกลุ่มข้อมูลแบบเคมีน

scikit-learn ถูกออกแบบสามารถใช้ร่วมกับไลบรารีนี้มไพและไซไพของภาษาไพธอนได้

Decision Tree

Decision Tree เป็น model แบบ rule-based คือ สร้างกฎ if-else จากค่าของแต่ละ feature โดยไม่มีสมการมากำกับความสัมพันธ์ระหว่าง feature & target สิ่งที่สำคัญในการสร้าง Decision Tree คือ การเลือก split ค่า feature แต่ละครั้ง จะต้อง minimise ค่าของ cost functionให้น้อยที่สุด

K-Nearest Neighbour Algorithm

K-Nearest Neighbour Algorithm เป็นวิธีที่ใช้ในการจัดแบ่งคลาส โดยเทคนิคนี้จะตัดสินใจว่า คลาสใดที่จะแทนเงื่อนไขหรือกรณีใหม่ๆ ได้บ้าง โดยการตรวจสอบจำนวนบางจำนวน (“K” ในขั้นตอนวิธีการเพื่อนบ้านใกล้ที่สุด) ของกรณีหรือเงื่อนไขที่เหมือนกันหรือใกล้เคียงกันมากที่สุด โดยจะหาผลรวม (Count Up) ของจำนวนเงื่อนไข หรือกรณีต่างๆ สำหรับแต่ละคลาส และกำหนดเงื่อนไขใหม่ๆ ให้คลาสที่เหมือนกันกับคลาสที่ใกล้เคียงกันมากที่สุด

Logistic Regression

- Logistic Regression เป็นเทคนิคทางสถิติภายใต้การดูแลเพื่อค้นหาความน่าจะเป็นของตัวแปรตาม (คลาสที่มีอยู่ในตัวแปร)
- โลจิสติกฟังก์ชันการใช้งานการถดถอยที่เรียกว่าฟังก์ชัน logit ที่จะช่วยให้ความสัมพันธ์การสืบทอดมาระหว่างตัวแปรตามและตัวแปรอิสระโดยการคาดคะเนความน่าจะเป็นหรือโอกาสของการเกิด
- ฟังก์ชันโลจิสติกส์ (หรือที่เรียกว่าฟังก์ชันซิกมอยด์) จะแปลงความน่าจะเป็นเป็นค่าไบนารีซึ่งสามารถใช้ในการคาดคะเนเพิ่มเติมได้

Naive Bayes

Naive Bayes classification เป็น Model หนึ่งใน การแบ่งกลุ่มที่เราต้องการโดยใช้ความน่าจะเป็นที่ชื่อว่า Naive bayes ยกตัวอย่างเช่น เราต้องการแบ่งกลุ่มว่าคนไข้ที่เข้ามานั้นเป็นไข้หวัดใหญ่หรือไม่ ซึ่งเราจะต้องถามอาการคนไข้มาให้ได้มากที่สุดว่าอาการเป็นอย่างไร แล้วเราถึงจะคาดคะเนจากข้อมูลอาการที่ได้ว่ามีความน่าจะเป็น ไข้หวัดใหญ่เท่าไร เช่นความน่าจะเป็น 95% เป็นต้น ซึ่งความน่าจะเป็นดังกล่าว เราจะใช้ Naive Bayes ในการหาค่าความน่าจะเป็นนั่นเอง

Neural Network

Neural Network หรือ Artificial Neural Network คือ โครงข่ายประสาทเทียม เป็นสาขาหนึ่งของปัญญาประดิษฐ์ Artificial Intelligence (AI) เป็นแนวคิดที่ออกแบบระบบโครงข่ายคอมพิวเตอร์ ให้เลียนแบบการทำงานของสมองมนุษย์

Random forest

Random forest เป็นหนึ่งในกลุ่มของโมเดลที่เรียกว่า Ensemble learning ที่มีหลักการคือการเทรนโมเดลที่เหมือนกันหลายๆ ครั้ง (หลาย Instance) บนข้อมูลชุดเดียวกัน โดยแต่ละครั้งของการเทรนจะเลือกส่วนของข้อมูลที่เทรนไม่เหมือนกัน แล้วเอาการตัดสินใจของโมเดลเหล่านั้นมาโหวตกันว่า Class ไหนถูกเลือกมากที่สุด

บทที่ 3

การทำงานของ code

```
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

1.1 import numpy as np เป็น Library มาตรฐานสำหรับการคำนวณและจัดการกับ Vector และ Matrix หลากมิติหรือเป็นอาเรย์

import pandas as pd เป็น Library สำหรับโหลดจัดการ และวิเคราะห์ข้อมูล Structured data เช่นตาราง csv

from sklearn.model_selection import train_test_split สำหรับการแบ่งข้อมูลออกเป็นสองกลุ่ม คือ train และ test

from sklearn.linear_model import LogisticRegression สำหรับการจำแนกประเภทข้อมูลที่ใช้ในการทำนายความน่าจะเป็นของตัวแปรตามประเภท

from sklearn.metrics import accuracy_score สำหรับการคำนวณคะแนนความแม่นยำของข้อมูล

```
# loading data
data = pd.read_csv('/content/pressure.csv')
```

1.2 โหลดข้อมูลเข้ามาเก็บไว้ในตัวแปร data

```
data.head(10)
```

1.3 แสดงข้อมูลจาก 10 ข้อมูลจากบนลงล่าง

	Age	Sex	Pulse	Pressure	Blood oxygen	Body temperature	target
0	78	1	80	120	99	36.5	0
1	67	1	75	126	100	36.8	0
2	82	1	98	180	98	36.7	3
3	69	0	93	169	98	36.2	2
4	85	1	69	110	99	35.9	0
5	80	0	79	132	100	37.0	0
6	73	0	73	198	99	36.1	3
7	77	1	82	174	100	36.0	2
8	90	1	89	188	99	36.6	3
9	66	0	73	98	99	35.8	0

```
data.shape
```

1.4 แสดงข้อมูลจากที่โหลดเข้ามาว่ามีทั้งหมดกี่แถวกี่คอลัมน์

```
# checking for missing values  
data.isnull().sum()
```

1.5 ตรวจสอบว่าข้อมูลที่โหลดเข้ามาเก็บไว้ในตัวแปร data มีข้อมูลที่หายไปหรือไม่

```
Age      0  
Sex      0  
Pulse    0  
Pressure 0  
Blood oxygen 0  
Body temperature 0  
target   0  
dtype: int64
```



```
# getting some info about the data
data.info()
```

1.6 ตรวจสอบว่าข้อมูลที่เก็บไว้ในตัวแปร data มีกี่คอลัมน์ และแต่ละคอลัมน์นั้นเป็นข้อมูลประเภทไหน

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 219 entries, 0 to 218
Data columns (total 7 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   Age             219 non-null   int64
1   Sex             219 non-null   int64
2   Pulse           219 non-null   int64
3   Pressure        219 non-null   int64
4   Blood oxygen    219 non-null   int64
5   Body temperature 219 non-null   float64
6   target          219 non-null   int64
dtypes: float64(1), int64(6)
memory usage: 12.1 KB
```

```
data.describe()
```

1.7 แสดงค่า count, mean, std, min, 25%, 50%, 75%, max ของแต่ละคอลัมน์

	Age	Sex	Pulse	Pressure	Blood oxygen	Body temperature	target
count	219.000000	219.000000	219.000000	219.000000	219.000000	219.000000	219.000000
mean	80.821918	0.465753	75.863014	149.616438	95.593607	36.438356	1.388128
std	9.977970	0.499969	14.910727	32.837226	3.152712	0.337542	1.075177
min	65.000000	0.000000	50.000000	91.000000	90.000000	35.700000	0.000000
25%	72.000000	0.000000	63.000000	123.000000	93.000000	36.200000	1.000000
50%	80.000000	0.000000	75.000000	145.000000	96.000000	36.400000	1.000000
75%	90.000000	1.000000	88.000000	177.000000	98.000000	36.700000	2.000000
max	99.000000	1.000000	100.000000	210.000000	100.000000	37.000000	3.000000

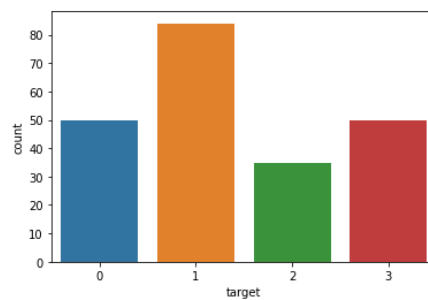
```
data['target'].value_counts()
```

1.8 แสดงข้อมูลในคอลัมน์ target ว่ามีข้อมูลอะไรบ้างและแต่ละข้อมูลมีอยู่จำนวนเท่าไร

```
1    84  
3    50  
0    50  
2    35  
Name: target, dtype: int64
```

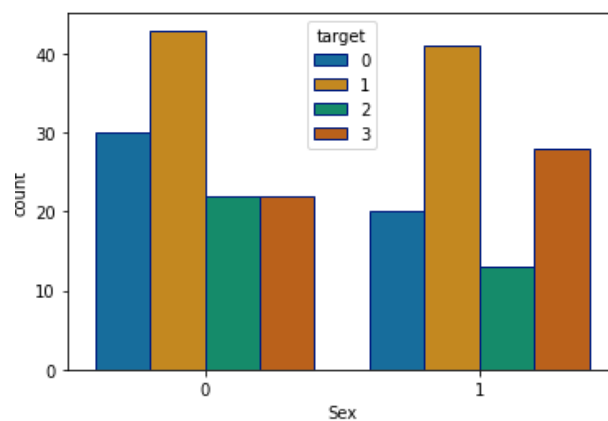
```
sns.countplot(data['target'])
```

1.9 สร้างกราฟจากข้อมูลในคอลัมน์ target



```
sns.countplot(x='Sex',hue='target',data=data,palette='colorblind',edgecolor=sns.color_palette('dark', n_colors=1))
```

1.10 สร้างกราฟจากข้อมูลในคอลัมน์ target และคอลัมน์ sex



```
X = data.drop(columns='target', axis=1)
Y = data['target']
print(X)
```

1.11 `X = data.drop(columns='target', axis=1)` คือให้ X เก็บค่าทุกคอลัมน์ ยกเว้นคอลัมน์ target

`Y = data['target']` คือให้ Y เก็บค่าคอลัมน์ target

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, stratify=Y, random_state=2)
```

1.12 แยกข้อมูลออกเป็นชุดแล้วทำการทดสอบแบบสุ่มตามจำนวนที่กำหนดไว้ โดยจาก code จะแบ่งข้อมูล ให้กับ test 30% และแบ่งให้ train 70%

```
model = LogisticRegression()
```

1.13 อัลกอริธึมการจำแนกประเภท Machine Learning ที่ใช้ในการทำนายความน่าจะเป็นของตัวแปรตามประเภท

```
model.fit(X_train, Y_train)
```

1.14 คือการหาความสัมพันธ์ของข้อมูลใน X_train และ Y_train

```
# training data
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
print('Accuracy on Training data : ', training_data_accuracy)
```

1.15 ทดสอบความแม่นยำของ train data

```
# test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
print('Accuracy on Test data : ', test_data_accuracy)
```

1.16 ทดสอบความแม่นยำของ test data

```
input_data = (80,0,98,120,98,36.5)
#input_data = (91,0,75,140,98,36.7)
#input_data = (72,0,98,165,92,35.9)
#input_data = (76,1,98,195,98,37)
input_array= np.asarray(input_data)

input_data_reshaped = input_array.reshape(1,-1)

prediction = model.predict(input_data_reshaped)
print(prediction)

if (prediction[0] == 0):
    print('ปกติ')
if (prediction[0]== 1):
    print('ผู้สูงอายุผิดปกติระดับที่ 1')
if (prediction[0]== 2):
    print('ผู้สูงอายุผิดปกติระดับ 2')
if (prediction[0]== 3):
    print('ผู้สูงอายุผิดปกติระดับ 3')
```

1.17 input_data คือข้อมูลที่ได้รับมาจากผู้ใช้

`input_array= np.asarray(input_data)` คือเปลี่ยนข้อมูลที่ได้รับจากผู้ใช้ให้เป็นอาเรย์

`input_data_reshaped = input_array.reshape(1,-1)` คือการเปลี่ยนรูปร่างใหม่ให้อาเรย์โดยไม่เปลี่ยนแปลงข้อมูล

`prediction = model.predict(input_data_reshaped)` คือการนำข้อมูลที่ผู้ใช้ใส่เข้ามาและทำการเปลี่ยนรูปเสร็จแล้วไปทำนายเพื่อหาคำตอบ