

智慧化企業整合
Project 3

利用 GaitPhase 資料進行步態速率分析

指導老師：
邱銘傳 教授

109034570 林溥鈞

目錄

第 1 章

研究主題說明

資料分析

DNN

2 【5W1H】

2 第 2 章 資料分析與前處理

2 資料來源

2 資料介紹

3 資料蒐集

3 資料合併

4

6 資料樣本區分

6 第 3 章 預測模型設定

6SVM

78 Logistic regression

錯誤! 尚未定義書籤。 XGBoost

7

8 第 4 章 模型效度驗證

9 準確率和訓練時間比較

9 混淆矩陣

9 泛化能力評估

9 不同身分別辨識

9 第 5 章 結論與未來展望

10 第 6 章 參考資料

11

第 1 章 研究主題說明

步態屬於人類生物特徵，也是良好的識別特徵，步態特徵具有獨特性、週期性和不變性這幾項重要的優勢，相比其他生物特徵像是人臉特徵，可能因為戴口罩或是化妝而造成識別上的困難。而步態分析就是利用這幾項優勢進行許多動作分析或是身分識別。而本研究想利用步態資料進行速率辨識，些微的速率改變無法簡單的利用肉眼看出資料的差異，所以想用 AI 的方式，找到快速且準確率高的模型進行辨識。而這研究可以幫助運動員進行訓練上的調整，病患的復健。本研究數據是透過 Description of the GaitPhase Database 這篇 paper 所取得。

【5W1H】

What?	利用步態特徵的獨特性、不變化，週期性分辨不同步態速率
Who?	可以用在運動員、復健病患、跌倒老人
When?	運動員在訓練時，病患在復健時
Where?	實驗室、訓練室、醫院
Why?	有些細微變化是難以分辨的，又或是必須借助儀器才能分析，希望可以透過 AI 的方式輔助分析，找出準確率最好的分類模型
How?	利用分類模型、深度學習、資料分析

第 2 章 資料分析與前處理

資料來源

本研究數據是透過 Description of the GaitPhase Database 這篇 paper 所提供的數據來訓練模型。

資料介紹

首先釐清各欄位資料所代表的資訊及其意涵，方便後續資料處理與模型分析得以順利進行，整理以下表格：

受測者	21位
性別	10位男性、11位女性
年齡	23.8 yrs \pm 3.3 yrs
身高	172.8 cm \pm 9.4 cm,
體重	66.6 kg \pm 10.9 kg
速度	[0.6, 1.7] m/s at 0.1 m/s

表一、受測者資料



圖一、資料收集實驗室

資料蒐集

利用 3D marker positions，在左腳和右腳各貼 4 個標記點，位置分別在蹠骨 1、蹠骨 2、蹠骨 5 和阿基里斯腱，總共有 24 個特徵值，分為左腳右腳、4 個標記點和三軸施力資料。

蹠骨



圖二、標記點

特徵24個 每一個有1200筆資料	
雙腳	左腳、右腳
Marker	1、2、5、Achilles tendon
3軸	x: posterior-anterior direction (前後方向)
	y: right-left direction (左右方向)
	z: inferior superior (vertical) direction (上下垂直方向)

表二、資料特徵介紹

資料合併

每個速率資料都是獨立 csv 檔，所以進行資料合併並分類命名 0、1

```
data1 = pd.DataFrame({'secim': np.zeros(12000)})
data2 = pd.DataFrame({'secim': np.ones(12000)})

data2 = pd.concat([dataseven2, data2], axis=1)
data1 = pd.concat([datasix1, data1], axis=1)

data = data1.append(data2, ignore_index=True)
data.head()
```

圖三、資料合併程式碼

資料分析

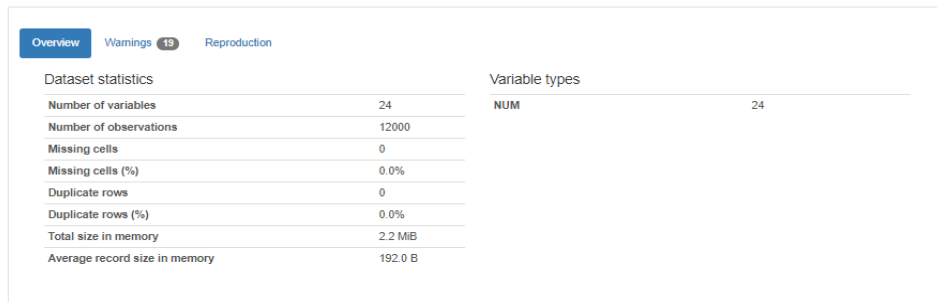
接著，將檔案以 dataframe 匯入，並利用 pandas 的

1. head()方法來檢查前幾項數據

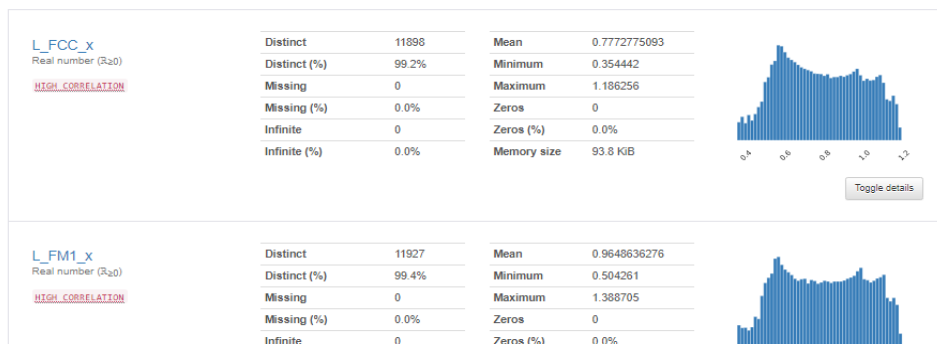
	L_FCC_x	L_FM1_x	L_FM2_x	L_FM5_x	R_FCC_x	R_FM1_x	R_FM2_x	R_FM5_x	L_FCC_y	L_FM1_y	L_FM2_y	L_FM5_y	R_FCC_y	R_FM1_y
0	0.628926	0.836563	0.840168	0.823058	0.888503	1.098850	1.094071	1.062056	0.561139	0.536024	0.584897	0.637451	0.439656	0.426320
1	0.626188	0.833782	0.837441	0.820210	0.894769	1.105314	1.100358	1.068544	0.561084	0.535987	0.584859	0.637403	0.439213	0.425575
2	0.623270	0.830821	0.834532	0.817181	0.901406	1.112139	1.106999	1.075402	0.561024	0.535950	0.584821	0.637356	0.438775	0.424830
3	0.620190	0.827697	0.831457	0.813991	0.908358	1.119264	1.113935	1.082572	0.560962	0.535914	0.584784	0.637311	0.438352	0.424102
4	0.616968	0.824431	0.828239	0.810662	0.915562	1.126620	1.121102	1.089989	0.560897	0.535879	0.584747	0.637268	0.437955	0.423410

2. `import pandas_profiling` 這是 pandas 內建的分析程式，可以快速整理所有資料，並多樣分析。

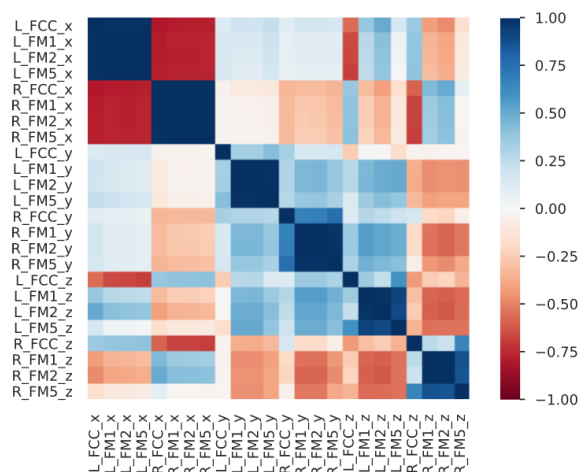
Overview



Variables



3. 透過 pandas_profiling 網頁資料分析，觀察關聯圖找到 5 個不顯著的特徵



圖四、關聯圖

不顯著特徵
L_FCC_y
R_FCC_y
L_FCC_z
R_FCC_z
L_FM5_z

表三、不顯著特徵

4. 進行特徵選取，選取剩下的 19 個特徵進行訓練測試，發現全特徵的訓練結果比較好。

模型	準確率	決策
SVM 24特徵	0.915	V
SVM 19特徵	0.761	

表四、全特徵和 19 特徵比較

資料樣本區分

以 sklearn 為資料選擇所提供的 train_test_split() 將資料分為特徵值與目標；特徵值為模型中的 x 值，目標值為模型中的 y 值，將資料分為訓練集與測試集，統一後續模型使用之訓練與測試資料。

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=1)
```

第 3 章 預測模型設定

模型實驗流程

使用相同受測者，不同速率進行資料訓練，這裡使用受測者 1 號，速率分別為 0.7、0.8 m/s。模型使用 SVM、Logistic、XGBoost、DNN 之後各別超參數優化，選出最好的模型並進行泛化能力測試。

SVM

SVM 模型進行 4 次超參數優化，有進行優化的超參數為 C 和 gamma，優化的數值和過程為圖五。

超參數	結果	超參數	結果	超參數	結果	超參數	結果
C=1	0.728	C=1	0.765	C=100	0.914	C=1000	0.988
kernel='rbf'		kernel='rbf'		kernel='rbf'		kernel='rbf'	
gamma='auto'		gamma='scale'		gamma='scale'		gamma='scale'	

acc_train of svm is : 0.9866071428571429
acc_test of svm is : 0.9880555555555556

圖五、SVM 超參數優化過程

Logistic

Logistic 模型進行 4 次超參數優化，有進行優化的超參數有 solver、C 和 max_iter，優化的數值和過程為圖六。

超參數	結果	超參數	結果	超參數	結果	超參數	結果
C=1	0.721	C=1000	0.838	C=1000	0.915	solver='liblinear'	0.915
solver='liblinear'		solver='liblinear'		solver='liblinear'		solver='sag'	0.871
max_iter=10		max_iter=10		max_iter=100		lbfgs	0.896
						newton-cg	0.914

train accuracy for Log Regressin is 0.9161309523809524
test accuracy for Log Regressin is 0.9155555555555556

圖六、Logistic 超參數優化過程

XGBoost

XGBoost 模型進行 4 次超參數優化，有進行優化的超參數有 learning_rate 和 max_depth，優化的數值和過程為圖七。

超參數	結果	超參數	結果	超參數	結果	超參數	結果
<u>learning_rate=0.1</u>	0.936	<u>learning_rate=0.3</u>	0.975	<u>learning_rate=0.2</u>	0.997	<u>learning_rate=0.2</u>	0.999
<u>max_depth=3</u>		<u>max_depth=3</u>		<u>max_depth=4</u>		<u>max_depth=5</u>	

acc_train of XGB is : 1.0
acc_test of XGB is : 0.9994444444444445

圖七、XGBoost 超參數優化過程

DNN

DNN 網路架構使用 5 個 Layers，activation 使用 relu，而最後的分類使用 softmax，而超參數優化選擇調整 epochs 和 batch_size，參數優化為圖九。

```

model = keras.Sequential([
    keras.layers.Dense(512, activation='relu'),
    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dense(64, activation='relu'),
    keras.layers.Dense(64, activation='relu'),
    keras.layers.Dense(10, activation='softmax')
])

#optimizer: descent algorithm
#loss: objective loss function
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

```

圖八、DNN 網路架構

超參數	結果	超參數	結果	超參數	結果
epochs=100	0.867	epochs=500	0.983	epochs=500	0.999
<u>batch_size=25</u>		<u>batch_size=25</u>		<u>batch_size=50</u>	

圖九、DNN 超參數優化

第 4 章 模型效度驗證

準確率和訓練時間比較

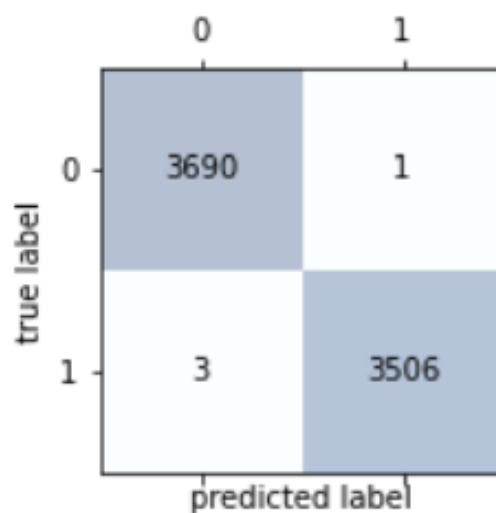
由表五可知，最好的模型選擇為 XGBoost

模型	準確率	訓練速度	決策
SVM	0.988	1分30秒	
Logistic	0.915	1分內	
<u>XGBoost</u>	0.999	1分內	✓
DNN	0.999	6分	

表五、模型效度比較

混淆矩陣

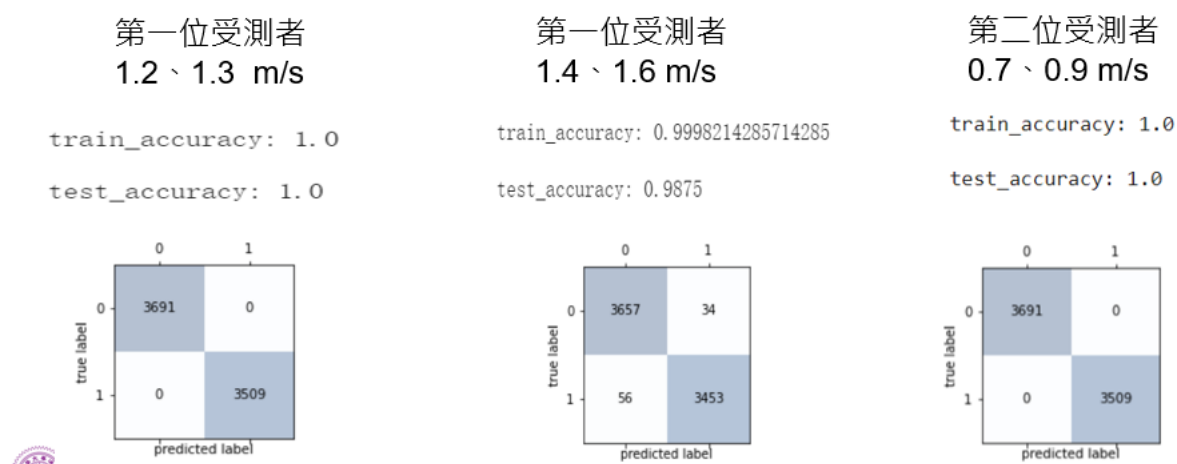
評估訓練出來的模型成效好不好，我們使用混淆矩陣，分辨模型在分類上的準確率。由圖十可以看出模型的分辨率效果非常好，型一和型二誤差非常小。



圖十、XGBoost 混淆矩陣

泛化能力評估

選擇三種資料進行泛化能力測試，由圖十一可以得知，每一組測試都沒有 overfitting，也沒有過多的型一和型二誤差。



圖十一、三組資料各別泛化能力

不同身分別辨識

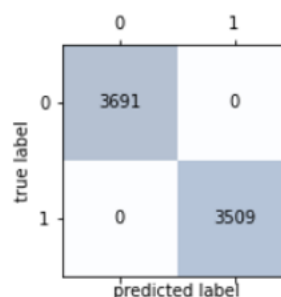
在進行泛化能力測試時，有使用相同速率但不同人的資料組合進行訓練，發現使用 XGBoost 成功辨識出不同的人效果也不錯，這可以做為未來新研究的依據。

- 第一位和第二位受測者
- 0.7 m/s
- 成功在相同速度下，辨識不同人

```
#第一位 0.7資料
datasix1 = pd.read_csv("GP1_0.7_marker.csv")
datasix1.head()
#第二位 0.7資料
dataseven2 = pd.read_csv("GP2_0.7_marker.csv")
dataseven2.head()
```

train_accuracy: 1.0

test_accuracy: 1.0



圖十二、相同速率但不同人的資料組合進行訓練

第 5 章 結論與未來展望

本篇研究成功使用分類模型、深度學習分辨出速率差異，更成功利用步態特徵的獨特性，在相同速率下辨識不同的人，其中 XGBoost 是個不錯的分類模型（準確率高、速度快）。目前可以在相同受測者下分辨不同速率，未來希望可以透過步態的週期性，結合時間序列型模型，去預測下一步的步態速率變化。

第 6 章 參考資料

<https://www.kaggle.com/dasmehdixtr/classiffier-examples-on-gait-phase-dataset>

<https://medium.com/jameslearningnote/%E8%B3%87%E6%96%99%E5%88%86%E6%9E%90-%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-%E7%AC%AC5-2%E8%AC%9B-kaggle%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92%E7%AB%B6%E8%B3%BD%E7%A5%9E%E5%99%A8xgboost%E4%BB%8B%E7%B4%B9-1c8f55cffcc>

<https://chih-sheng-huang821.medium.com/%E6%A9%9F%E5%99%A8-%E7%B5%B1%E8%A8%88%E5%AD%B8%E7%BF%92-%E7%BE%85%E5%90%89%E6%96%AF%E5%9B%9E%E6%AD%B8-logic-regression-aff7a830fb5d>

<https://t.codebug.vip/questions-1743248.htm>