

# Machine Learning Final Project

0416303 楊博凱 0416323 黃右萱 0416305 王柏堯

## Scheme

---

特徵抽取→預測模型→綜合判斷

## 資料來源

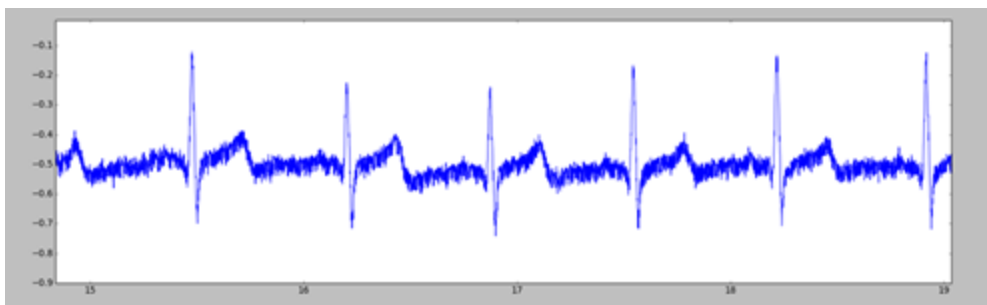
---

訊號與系統實驗專題的資料，經過 IRB 認證

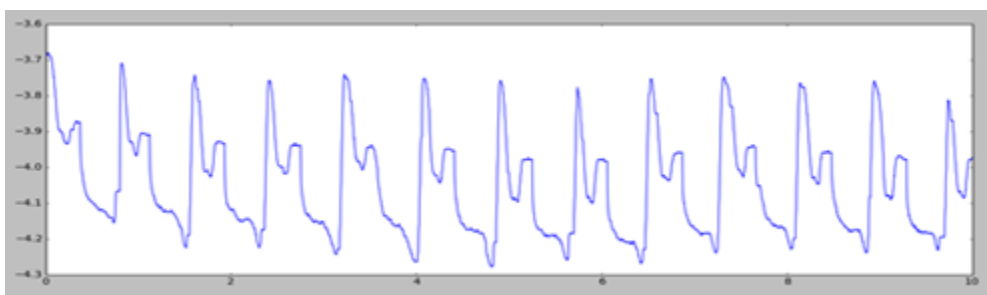
在身上配戴 ECG，脈搏，胸與腹的大小的測量裝置，測量觀看恐怖的影片與興奮的影片，以及平靜時的身體狀態，為了簡化計算量，我們就將這些訊號歸類為 1.害怕 2.興奮 3.平靜 來當作 target。

總共收集到 5 位同學願意協助給予資料，本次專案將專注於 ECG 與脈搏之分析。

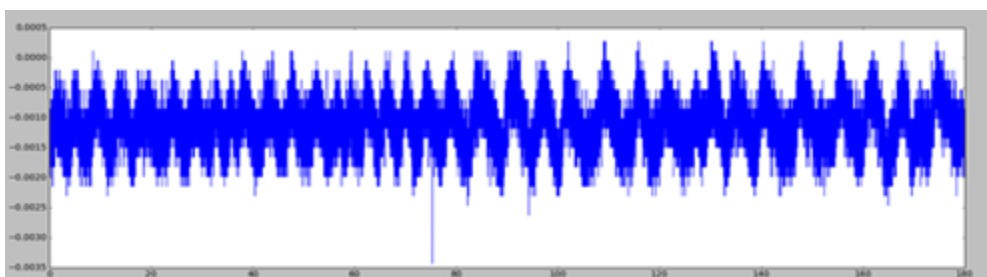
- ECG



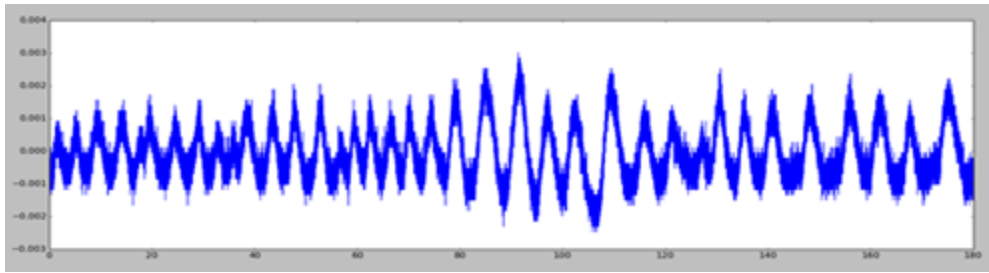
- Blood Pulse



- Chest



- Abdomen

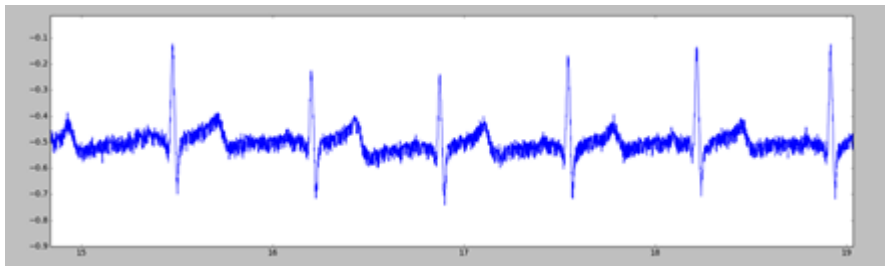


## 特徵抽取

---

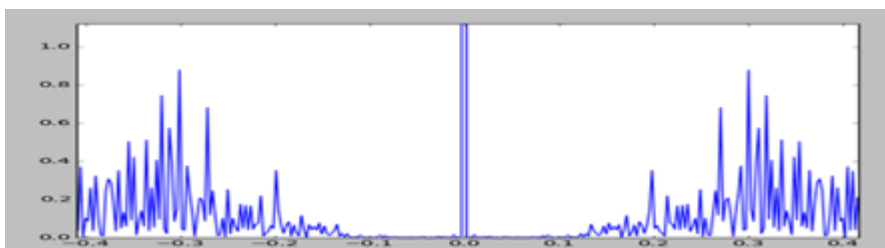
### Naive

- 對時間做切割，直接作為 descriptive feature



### Fourier Transform

- 對時間做若干切割，然後做快速傅立葉轉換，取其 Amplitude Spectrum 作為 descriptive Feature



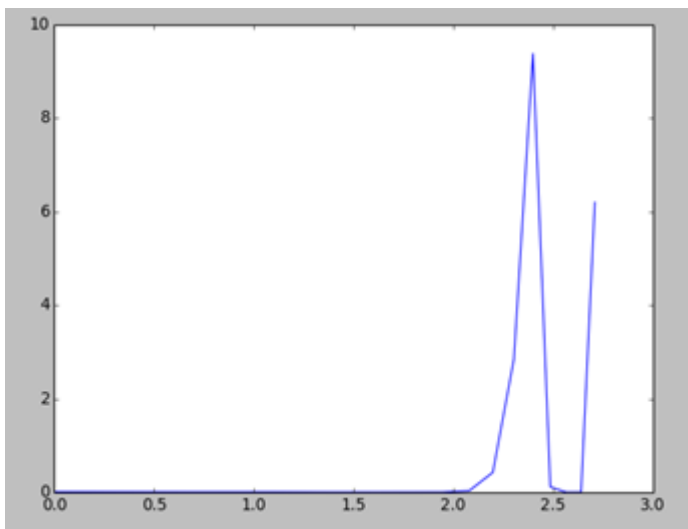
### Empirical Mode Decomposition (EMD)

- Goal :
  - 找出非穩態訊號的不同趨勢訊號
  - 找出滿足 Hilbert 假設的趨勢訊號
    - 零均值
    - 局部對稱

- Scheme:
  - 迭代找出若干個 Intrinsic Mode Function(IMF)
    - 找平均包絡線
      - 上下包絡線之平均
      - 包絡線由對區域極值找 Cubic Spline(RK4)求得
    - 訊號鉗位
      - 當前訊號扣除平均包絡線
      - 直到滿足終止條件(可以訊號標準差等界定)
  - 找出 IMF 後將原本訊號扣除 IMF 得到 Residual Signal
    - 若 Residual Signal 為大致單調，則結束 EMD
    - 否則對 Residual Signal 求 IMF

## Power Analysis

- Concept：找出重要的 IMF 分量進行分析
- 如下圖，各個 x 值對應到一個 IMF，我們將挑選出 peaked power IMF(s)



## Down Sampling:

- Goal:
  - 增加可用 Feature 個數
  - 加速 EMD 拆解速度
    - 實作上原本數分鐘的處理能在秒至毫秒內完成
  - Concept：
    - 將訊號拆解成若干較低取樣精度的訊號
  - Description：

給定序列  $\langle a_k \rangle, k \in [0, nm)$

取分割子序列集  $\{\langle b_j^i \rangle : i \in [0, n)\}$  where  $b_j^i = a_{j \cdot m + i}$

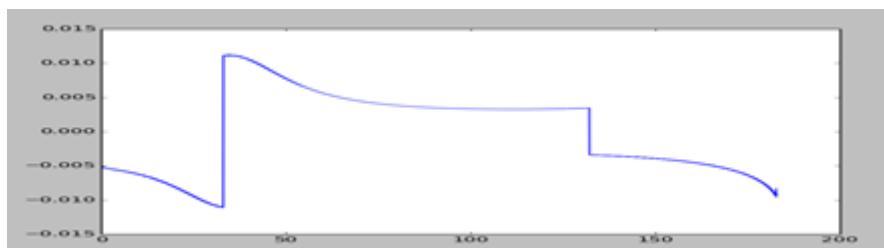
## Hilbert Huang Transform (Modified for Feature Extraction)

- Hilbert Huang Transform :
  - Scheme :
    - Down Sampling
    - EMD
    - Power Analysis
    - Hilbert Transform
    - Obtain Instantaneous Phase & Frequency (IF)
    - Down Sampling
  - Hilbert Transform :

$$H\{u\}(t) = p.v. \int_{-\infty}^{\infty} \frac{u(\tau)}{\pi \cdot (t - \tau)} d\tau$$

- From Hilbert to IF :

- $\bar{u}(t) = H\{u\}(t)$
- $s(t) = u(t) + j \cdot \bar{u}(t)$
- $s(t) = m(t)e^{j\theta(t)}$ , for real valued  $m(t), \theta(t)$ 
  - Phase Determination (Multiple Ways):
    - use arctan transformation
    - use absolute value
- $f(t) = \frac{\omega(t)}{2\pi}$  where  $\omega(t) = \frac{d\theta(t)}{dt}$



## Direct Quadrature (Modified for Feature Extraction)

- Direct Quadrature :
  - Scheme :
    - Down Samplin
    - EMD
    - Power Analysis
    - Normalize

- Obtain Instantaneous Phase & Frequency (IF)
- Down Sampling
- Concept :
  - 對訊號標準化，並將之視為弦波函數取 IF
- Normalize :
  - 找出 IMF 絕對極值
  - 以 RK4 找出其上包絡線
  - 將整個訊號除去上包絡線
- 找 IF :



- 考慮信號 $u(t)$ 標準化後為 $\bar{u}(t)$
- 取出瞬時相位： $\theta(t) = \tan^{-1}\left(\frac{\sqrt{1-\bar{u}^2(t)}}{\bar{u}(t)}\right)$
- 得到瞬時頻率： $f(t) = \frac{d\theta(t)}{dt}$

## Learning Model

---

- 決策樹 (Decision Tree) : 我們採用 Random Forest 對多個 Decision Tree 的結論做出綜合決策
  - n\_estimators (Number of trees in the forest) = 5
  - max\_depth (Maximum depth of the tree) = 3
- 支持向量機 (Support Vector Machine) : 我們運用 sklearn.svm.SVC 來實作 SVM
- 高斯機率模型 (Gaussian Naive Bayssian) : 我們運用 sklearn.naive\_bayes.GaussianNB 來實作 Probability
- 類神經網絡 (Neural Network) : 我們運用 sklearn.neural\_network.MLPClassifier 來實作 NN
  - Solver (Solver for weight optimization) = 'lbfgs' (an optimizer of quasi-Newton methods)
  - Activation (Activation function for the hidden layer) = 'tanh' (returns  $f(x) = \tanh(x)$ )
- K-近鄰算法 (K-Nearest Neighbor) : 我們運用 sklearn.neighbors.KNeighborsClassifier 來實作 KNN
  - n\_neighbors (Number of neighbors) = 10
  - metric (Distance metric to use for the tree) = 'manhattan'
  - weights = 'distance' (Closer neighbors of a point will have greater influence than those which are far away)

最後，實際使用時，我們製作了一個模組，傳入生理信息，將自動抽取出特徵，並做多次預測，最後以 voting 的方式決定最後預測結果。

## Validation Result

---

### 單人數據交叉審驗

- 採用同一人的生理信號，搗亂後分割為訓練以及測試數據集
- 10 折交叉審驗
- ECG

```
Mode: naive
Extract Data from sub004_baseline_ECG.csv
Extract Data from sub004_film_2_ECG.csv
Extract Data from sub004_film_1_ECG.csv

DT Train...
[[ 0 13 5]
 [ 0 53 26]
 [ 0 52 26]]
0.451428571429

SVM Train...
[[ 0 18 0]
 [ 0 79 0]
 [ 0 78 0]]
0.451428571429

Prob Train...
[[ 4 7 7]
 [16 28 35]
 [10 32 36]]
0.388571428571

NN Train...
[[ 0 11 7]
 [ 0 40 39]
 [ 0 37 41]]
0.462857142857

KNN Train...
[[ 0 7 11]
 [ 1 43 35]
 [ 0 24 54]]
0.554285714286
```

```
Mode: fourier
Extract Data from sub004_baseline_ECG.csv
Extract Data from sub004_film_2_ECG.csv
Extract Data from sub004_film_1_ECG.csv

DT Train...
[[ 0 10 8]
 [ 0 65 14]
 [ 0 43 35]]
0.571428571429

SVM Train...
[[ 0 14 4]
 [ 0 55 24]
 [ 0 17 61]]
0.662857142857

Prob Train...
[[13 4 1]
 [ 2 70 7]
 [ 4 15 59]]
0.811428571429

NN Train...
[[ 3 9 6]
 [ 8 51 20]
 [ 2 12 64]]
0.674285714286

KNN Train...
[[ 0 12 6]
 [ 0 49 30]
 [ 0 16 62]]
0.634285714286
```

```
Mode: hilbert
Extract Data from sub004_baseline_ECG.csv
Extract Data from sub004_film_2_ECG.csv
Extract Data from sub004_film_1_ECG.csv

DT Train...
[[18 0 0]
 [ 0 78 0]
 [ 0 0 78]]
1.0

SVM Train...
[[ 0 0 18]
 [ 0 0 78]
 [ 0 0 78]]
0.448275862069

Prob Train...
[[18 0 0]
 [ 0 74 4]
 [ 0 11 67]]
0.913793103448

NN Train...
[[18 0 0]
 [ 0 78 0]
 [ 0 0 78]]
1.0

KNN Train...
[[18 0 0]
 [ 0 78 0]
 [ 0 0 78]]
1.0
```

```
Mode: DQ
Extract Data from sub004_baseline_ECG.csv
Extract Data from sub004_film_2_ECG.csv
Extract Data from sub004_film_1_ECG.csv

DT Train...
[[18 0 0]
 [ 0 78 0]
 [ 0 0 78]]
1.0

SVM Train...
[[ 0 0 18]
 [ 0 5 73]
 [ 0 2 76]]
0.465517241379

Prob Train...
[[18 0 0]
 [ 1 73 4]
 [ 3 2 73]]
0.942528735632

NN Train...
[[18 0 0]
 [ 0 77 1]
 [ 0 0 78]]
0.994252873563

KNN Train...
[[18 0 0]
 [ 0 78 0]
 [ 0 0 78]]
1.0
```

- 脈搏

```
Mode: naive
Extract Data from sub004_baseline_BloodPulse.csv
Extract Data from sub004_film_2_BloodPulse.csv
Extract Data from sub004_film_1_BloodPulse.csv

DT Train...
[[ 0 9 9]
 [ 0 57 22]
 [ 0 16 62]]
0.68

SVM Train...
[[ 0 17 1]
 [ 0 66 13]
 [ 0 39 39]]
0.6

Prob Train...
[[16 0 2]
 [ 0 74 5]
 [37 1 40]]
0.742857142857

NN Train...
[[ 4 9 5]
 [ 3 72 4]
 [ 0 0 78]]
0.88

KNN Train...
[[ 9 0 9]
 [ 0 74 5]
 [ 0 0 78]]
0.92
```

```
Mode: fourier
Extract Data from sub004_baseline_BloodPulse.csv
Extract Data from sub004_film_2_BloodPulse.csv
Extract Data from sub004_film_1_BloodPulse.csv

DT Train...
[[ 0 8 10]
 [ 0 74 5]
 [ 0 7 71]]
0.828571428571

SVM Train...
[[14 0 4]
 [ 0 79 0]
 [ 0 0 78]]
0.977142857143

Prob Train...
[[ 3 3 12]
 [19 32 28]
 [ 9 7 62]]
0.554285714286

NN Train...
[[ 0 18 0]
 [ 0 79 0]
 [ 0 78 0]]
0.451428571429

KNN Train...
[[11 0 7]
 [ 0 75 4]
 [ 0 0 78]]
0.937142857143
```

```
Mode: hilbert
Extract Data from sub004_baseline_BloodPulse.csv
Extract Data from sub004_film_2_BloodPulse.csv
Extract Data from sub004_film_1_BloodPulse.csv

DT Train...
[[18 0 0]
 [ 0 78 0]
 [ 0 0 78]]
1.0

SVM Train...
[[ 0 17 1]
 [ 0 78 0]
 [ 0 72 6]]
0.48275862069

Prob Train...
[[18 0 0]
 [ 0 57 21]
 [ 0 12 66]]
0.810344827586

NN Train...
[[18 0 0]
 [ 0 78 0]
 [ 0 0 78]]
1.0

KNN Train...
[[18 0 0]
 [ 0 78 0]
 [ 0 0 78]]
1.0
```

```
Mode: DQ
Extract Data from sub004_baseline_BloodPulse.csv
Extract Data from sub004_film_2_BloodPulse.csv
Extract Data from sub004_film_1_BloodPulse.csv

DT Train...
[[18 0 0]
 [ 0 78 0]
 [ 0 0 78]]
1.0

SVM Train...
[[ 0 13 5]
 [ 0 2 76]
 [ 0 0 78]]
0.459770114943

Prob Train...
[[18 0 0]
 [ 0 77 1]
 [ 0 0 78]]
0.994252873563

NN Train...
[[18 0 0]
 [ 0 78 0]
 [ 0 0 78]]
1.0

KNN Train...
[[18 0 0]
 [ 0 78 0]
 [ 0 0 78]]
1.0
```

## 多人數據交叉審驗

- 採用多人的生理數據，搗亂後分割為訓練以及測試數據集
- 10 折交叉審驗



- ECG

```
Mode: naive
Extract Data from sub017_baseline_ECG.csv
Extract Data from sub017_film_2_ECG.csv
Extract Data from sub017_film_1_ECG.csv
Extract Data from sub018_baseline_ECG.csv
Extract Data from sub018_film_2_ECG.csv
Extract Data from sub018_film_1_ECG.csv
Extract Data from sub022_baseline_ECG.csv
Extract Data from sub022_film_2_ECG.csv
Extract Data from sub022_film_1_ECG.csv
Extract Data from sub004_baseline_ECG.csv
Extract Data from sub004_film_2_ECG.csv
Extract Data from sub004_film_1_ECG.csv

DT Train...
[[ 0 46 26]
 [ 0 224 88]
 [ 0 192 117]]
0.492063492063

SVM Train...
[[ 0 55 17]
 [ 0 248 64]
 [ 0 237 72]]
0.46176046176

Prob Train...
[[ 16 30 26]
 [ 99 151 102]
 [ 77 132 100]]
0.385281385281

NN Train...
[[ 0 34 38]
 [ 1 174 137]
 [ 3 152 154]]
0.473304473304

KNN Train...
[[ 2 35 35]
 [ 1 198 113]
 [ 3 132 174]]
0.539682539682
```

```
Mode: fourier
Extract Data from sub017_baseline_ECG.csv
Extract Data from sub017_film_2_ECG.csv
Extract Data from sub017_film_1_ECG.csv
Extract Data from sub018_baseline_ECG.csv
Extract Data from sub018_film_2_ECG.csv
Extract Data from sub018_film_1_ECG.csv
Extract Data from sub022_baseline_ECG.csv
Extract Data from sub022_film_2_ECG.csv
Extract Data from sub022_film_1_ECG.csv
Extract Data from sub004_baseline_ECG.csv
Extract Data from sub004_film_2_ECG.csv
Extract Data from sub004_film_1_ECG.csv

DT Train...
[[ 0 37 35]
 [ 0 211 101]
 [ 0 190 119]]
0.47619047619

SVM Train...
[[ 2 39 31]
 [ 1 243 68]
 [ 1 84 224]]
0.676767676768

Prob Train...
[[ 23 33 16]
 [ 36 204 72]
 [ 22 134 153]]
0.548340548341

NN Train...
[[ 0 20 52]
 [ 0 161 151]
 [ 0 103 206]]
0.529581529582

KNN Train...
[[ 9 29 34]
 [ 1 208 103]
 [ 2 98 209]]
0.614718614719
```

```
Mode: hilbert
Extract Data from sub017_baseline_ECG.csv
Extract Data from sub017_film_2_ECG.csv
Extract Data from sub017_film_1_ECG.csv
Extract Data from sub018_baseline_ECG.csv
Extract Data from sub018_film_2_ECG.csv
Extract Data from sub018_film_1_ECG.csv
Extract Data from sub022_baseline_ECG.csv
Extract Data from sub022_film_2_ECG.csv
Extract Data from sub022_film_1_ECG.csv
Extract Data from sub004_baseline_ECG.csv
Extract Data from sub004_film_2_ECG.csv
Extract Data from sub004_film_1_ECG.csv

DT Train...
[[ 72 0 0]
 [ 0 180 128]
 [ 0 33 275]]
0.765988372893

SVM Train...
[[ 0 0 72]
 [ 0 1 307]
 [ 0 0 308]]
0.449127906977

Prob Train...
[[ 72 0 0]
 [ 40 266 32]
 [ 0 207 93]]
0.626453488372

NN Train...
[[ 72 0 0]
 [ 0 266 42]
 [ 0 54 254]]
0.86046511679

KNN Train...
[[ 72 0 0]
 [ 0 306 2]
 [ 0 0 308]]
0.997093023256
```

```
Mode: DQ
Extract Data from sub017_baseline_ECG.csv
Extract Data from sub017_film_2_ECG.csv
Extract Data from sub017_film_1_ECG.csv
Extract Data from sub018_baseline_ECG.csv
Extract Data from sub018_film_2_ECG.csv
Extract Data from sub018_film_1_ECG.csv
Extract Data from sub022_baseline_ECG.csv
Extract Data from sub022_film_2_ECG.csv
Extract Data from sub022_film_1_ECG.csv
Extract Data from sub004_baseline_ECG.csv
Extract Data from sub004_film_2_ECG.csv
Extract Data from sub004_film_1_ECG.csv

DT Train...
[[ 68 0 6]
 [ 0 212 96]
 [ 0 40 268]]
0.793604651163

SVM Train...
[[ 0 11 61]
 [ 0 0 302]
 [ 0 6 302]]
0.44674418605

Prob Train...
[[ 72 0 0]
 [ 16 278 41]
 [ 24 178 106]]
0.662790697674

NN Train...
[[ 69 1 2]
 [ 0 246 62]
 [ 1 50 257]]
0.891395340837

KNN Train...
[[ 72 0 0]
 [ 0 301 7]
 [ 0 0 300]]
0.978197674419
```

- 脈搏

```
Mode: naive
Extract Data from sub017_baseline_BloodPulse.csv
Extract Data from sub017_film_2_BloodPulse.csv
Extract Data from sub017_film_1_BloodPulse.csv
Extract Data from sub018_baseline_BloodPulse.csv
Extract Data from sub018_film_2_BloodPulse.csv
Extract Data from sub018_film_1_BloodPulse.csv
Extract Data from sub022_baseline_BloodPulse.csv
Extract Data from sub022_film_2_BloodPulse.csv
Extract Data from sub022_film_1_BloodPulse.csv
Extract Data from sub004_baseline_BloodPulse.csv
Extract Data from sub004_film_2_BloodPulse.csv
Extract Data from sub004_film_1_BloodPulse.csv

DT Train...
[[ 0 0 72]
 [ 0 107 205]
 [ 0 12 297]]
0.582972582973

SVM Train...
[[ 0 41 31]
 [ 0 157 155]
 [ 0 126 183]]
0.49062049062

Prob Train...
[[ 90 2 20]
 [ 86 116 110]
 [ 94 37 178]]
0.496392496392

NN Train...
[[ 0 4 60]
 [ 0 119 193]
 [ 0 13 296]]
0.598845598846

KNN Train...
[[ 27 17 28]
 [ 6 262 44]
 [ 5 34 270]]
0.806637806638
```

```
Mode: fourier
Extract Data from sub017_baseline_BloodPulse.csv
Extract Data from sub017_film_2_BloodPulse.csv
Extract Data from sub017_film_1_BloodPulse.csv
Extract Data from sub018_baseline_BloodPulse.csv
Extract Data from sub018_film_2_BloodPulse.csv
Extract Data from sub018_film_1_BloodPulse.csv
Extract Data from sub022_baseline_BloodPulse.csv
Extract Data from sub022_film_2_BloodPulse.csv
Extract Data from sub022_film_1_BloodPulse.csv
Extract Data from sub004_baseline_BloodPulse.csv
Extract Data from sub004_film_2_BloodPulse.csv
Extract Data from sub004_film_1_BloodPulse.csv

DT Train...
[[ 0 20 52]
 [ 0 192 120]
 [ 0 68 241]]
0.62481962482

SVM Train...
[[ 39 19 14]
 [ 9 272 31]
 [ 11 18 280]]
0.852813852814

Prob Train...
[[ 6 11 95]
 [ 14 71 227]
 [ 23 45 241]]
0.458874458874

NN Train...
[[ 0 7 65]
 [ 0 146 166]
 [ 0 7 302]]
0.646464646465

KNN Train...
[[ 39 17 16]
 [ 7 259 46]
 [ 5 25 279]]
0.832611832612
```

```
Mode: hilbert
Extract Data from sub017_baseline_BloodPulse.csv
Extract Data from sub017_film_2_BloodPulse.csv
Extract Data from sub017_film_1_BloodPulse.csv
Extract Data from sub018_baseline_BloodPulse.csv
Extract Data from sub018_film_2_BloodPulse.csv
Extract Data from sub018_film_1_BloodPulse.csv
Extract Data from sub022_baseline_BloodPulse.csv
Extract Data from sub022_film_2_BloodPulse.csv
Extract Data from sub022_film_1_BloodPulse.csv
Extract Data from sub004_baseline_BloodPulse.csv
Extract Data from sub004_film_2_BloodPulse.csv
Extract Data from sub004_film_1_BloodPulse.csv

DT Train...
[[ 67 0 5]
 [ 0 281 27]
 [ 0 132 176]]
0.761627906977

SVM Train...
[[ 0 25 47]
 [ 0 11 207]
 [ 0 1 307]]
0.462209302326

Prob Train...
[[ 70 0 2]
 [ 0 286 22]
 [ 0 140 168]]
0.761627906977

NN Train...
[[ 71 0 1]
 [ 1 280 27]
 [ 0 9 299]]
0.94476744186

KNN Train...
[[ 72 0 0]
 [ 0 308 0]
 [ 0 0 308]]
1.0
```

```
Mode: DQ
Extract Data from sub017_baseline_BloodPulse.csv
Extract Data from sub017_film_2_BloodPulse.csv
Extract Data from sub017_film_1_BloodPulse.csv
Extract Data from sub018_baseline_BloodPulse.csv
Extract Data from sub018_film_2_BloodPulse.csv
Extract Data from sub018_film_1_BloodPulse.csv
Extract Data from sub022_baseline_BloodPulse.csv
Extract Data from sub022_film_2_BloodPulse.csv
Extract Data from sub022_film_1_BloodPulse.csv
Extract Data from sub004_baseline_BloodPulse.csv
Extract Data from sub004_film_2_BloodPulse.csv
Extract Data from sub004_film_1_BloodPulse.csv

DT Train...
[[ 65 5 2]
 [ 0 262 46]
 [ 0 110 198]]
0.763081395349

SVM Train...
[[ 3 48 21]
 [ 0 301 7]
 [ 0 231 77]]
0.553779069767

Prob Train...
[[ 67 5 0]
 [ 1 290 17]
 [ 1 102 205]]
0.816860465116

NN Train...
[[ 71 1 0]
 [ 0 305 3]
 [ 0 3 305]]
0.989825581395

KNN Train...
[[ 72 0 0]
 [ 0 308 0]
 [ 0 0 308]]
1.0
```

## 多人數據分割審驗

- 採用多人的生理數據訓練，並採用另外一人作為測試數據
- 此測試皆採用平靜狀態之數據做為測試



- ECG

[illegible][illegible][illegible]

```
* * * * *
```

```
Predict
```

```
* * * * *
```

```
Extract Data from sub019_baseline_ECG.csv
```

```
wlum 1001
```

```
Down Sampling...
```

```
Energy Analyzing...
```

```
Instantaneous Frequency Computing...
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
2 2 2 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ]
```

```
DT : Normal
```

```
[1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 2 2 2  
2 2 2 2 2 2 2 2 2 0 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1  
1 1 1 1 1 2 2 1 1 1 1 1 1 1 2 2 1 1 1 1 1 2 2 2 2 1 1 1 1 1 1 1 2 1 1  
1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 2 2 2 2 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1  
1 1 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 1 1 1 1 1 1 ]
```

```
SVM : Excited
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ]
```

```
Prob : Normal
```

```
[2 0 0 0 0 0 0 2 2 2 0 0 0 0 0 0 0 2 2 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1 0  
0 0 0 0 0 0 0 0 2 2 1 2 2 1 1 1 1 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0  
0 0 2 0 0 0 0 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 0 0 0 0 0 0 0 0 2  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 2 2 2 1  
1 2 1 2 1 2 1 2 1 2 2 2 2 2 1 2 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 ]
```

```
NN : Normal
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 2 2 0 0 2 2 2 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 2 1 2 2 2 2 2 2 2 2 2 0 0 0 0 0 0 2 2 0 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
0 0 0 2 2 1 2 0 0 0 2 2 2 2 0 1 0 2 2 2 2 1 0 0 2 2 2 2 2 2 2 ]
```

```
KNN : Normal
```

● 脈搏

[illegible][illegible]



## 結論

---

## Feature Extraction

人體是非穩態系統，並不會滿足 Fourier 的週期性假設，所以想要單純使用傅立葉分析是不切實際的；再者，原訊號中存在許多測量誤差，以及干擾信號，所以也不應該是 naive；取出重要信息分量，而後對於當下的頻率去做分析才是合理的，所以 HHT 與 DQ 取出的瞬時頻率會有最好的結果。

## Model

- DT 能有良好的結果，推測是因為 target 只有做簡單的分類，所以在選擇上較不會出錯
- SVM 的結果不好可能是這些訊號並沒有辦法用超平面去切成三區不同的類別
- probability 的結果為普通，可能因為 naive 的 model 又把一些誤差帶回來，使得運算加快，但準確率降低
- NN 能夠處理複雜的 feature，所以這次表現算是優良，但因為有多層神經網絡的關係，使得運算非常緩慢
- KNN 效果非常良好，推測原因為特徵空間中同狀態分類的訊號相似程度高

總結來說，我們要先用 HHT 或 DQ 處理訊號，使得準確率大幅上升，再配合 KNN 或 DT 來當作 model(速度較快)，但若要進一步使用，當 target 變多變複雜時，就要使用 NN 了。

\*\*\* 由於我們的資料部分無法公開，所以部分資料將不放到 zip 檔中 \*\*\*

\*\*\*\*\* 組員分工 \*\*\*\*\*

Learning Model & PPT : 0416303 楊博凱

Feature Extraction & PPT : 0416323 黃右萱

兩邊協調 & 上台報告 : 0416305 王柏堯