

# 中華民國第 58 屆中小學科學展覽會 作品說明書

---

高級中等學校組 電腦與資訊學科

佳作

探究精神獎

052503

以機器學習模型為主旋律配置和弦

學校名稱：臺北市立建國高級中學

作者： 高二 張鑑璿	指導老師： 許雅淳
---------------	--------------

關鍵詞：和弦、機器學習、樂理

## 摘要

本研究訓練出機器學習中的隨機森林模型，以流行音樂的主旋律為輸入，輸出適合該段主旋律的和弦，而模型預測結果分布中得到的和弦功能與相互關係和樂理中的和弦功能理論大致吻合。在輸入的特徵值沒有使用連續性的樂句的條件下，本研究也分析了使用所謂「循環代碼」對於提高模型預測正確率的貢獻度，結果是有顯著的正面影響。本研究所聚焦的「配和弦」具體化創作構想（主旋律）的功能，因此與歌曲分析或音樂生成大不相同、而具有電腦輔助創作的獨特研究價值。

## 壹、研究動機

我從四歲的時候開始學習古典鋼琴，不過一直到了我升上高中、加入爵士音樂社之後，才開始接觸到流行與爵士音樂，也才開始去探究關於節拍、調式、和弦的進階樂理。在試著完整的呈現一首流行或爵士的曲子時，除了旋律要完全正確外，伴奏也是一樣。而伴奏完全正確在一般的情況下，指的就是使用相同的和弦（下文會有名詞介紹）伴奏。在原曲裡面通常會有不同的樂器把這些音符呈現出來，也因此有不少相關的研究是透過一首完整的歌的聲音檔案去分析當中的和弦。然而這些歌一開始是怎麼決定用（也就是音樂上常說的「配和弦」）那些和弦的？經過一些搜尋後，我發現這方面的研究幾乎不存在；實際操作方面，我曾經配過無數次的和弦，卻找不出一個特定、可供遵循的模式。因此在有主旋律、而沒有伴奏的情況下，如何利用旋律「無中生有」伴奏的和弦，便成為了我研究的主題。

## 貳、研究目的

我希望建構與訓練機器學習模型，以主旋律為唯一的輸入，輸出該段主旋律的和弦進行。研究動機中有提到一些已存在的研究，但它們偏向於分析已存在的事實，這裡提到的配和弦卻是強化、具體化大綱（主旋律）的工作，因此具有電腦輔助創作上，與歌曲分析或音樂生成大不相同的獨特研究價值，也是最能與真人創作結合的一種應用。而透過模型在輸出和弦的過程中，容易與正確答案混淆的和弦之分布，也可以尋找有關和弦功能的規律。透過控制訓練模型時取用的特徵值，也可以比較不同的因素對於和弦配置影響的大小。

## 參、研究設備與器材

### 一、家中的桌上型電腦，規格如下

- (一) 處理器：Intel® Xeon® CPU E3-1230 V2 @ 3.30GHz
- (二) 記憶體：16.0 GB
- (三) 儲存空間：SSD 475 GB
- (四) 作業系統：Windows 10 專業版，版本號 1709

### 二、軟體與網站

- (一) YouTube.com：有推薦播放清單，因此可以提供大量的歌曲來源
- (二) MuseScore.com：是除了 Wikifonia 之外的另一個樂譜資料來源，但品質較不穩定。
- (三) Wikifonia：由中研院 MAC (Music and Audio Computing Lab) 楊奕軒博士所分享的資料庫，現已因版權續約問題而失效，故無法直接透過網路查詢，是含有主旋律與和弦符號的 Music XML 檔案。
- (四) MuseScore2 (2.1.0) 一款免費的樂譜編輯軟體，有自己的檔案格式，但也可以開啟壓縮過的 Music XML 檔案與一般的 XML 檔案，並進行編輯與轉檔。
- (五) Notepad++ (6.2.3.0)：一款可以編輯多種文字格式的編輯器。
- (六) Python (3.6.4)：一個程式語言，網路上有許多機器學習的函式庫。
- (七) Spyder (3.2.6)：一個 Python 程式語言的開發、編譯與執行的整合環境。
- (八) scikit-learn (0.19.1)：一個以 Python 語言為基礎的機器學習函式庫。
- (九) Microsoft Onedrive.com：微軟所創之雲端硬碟，供資料同步使用。
- (十) Microsoft Excel 2013：用以整理、分析數據與製作圖表。
- (十一) Microsoft Word 2013：用以撰寫科學展覽作品說明書。

## 肆、研究過程與方法

### 一、文獻探討

#### (一) 本研究中使用到的樂理

##### 1. 和弦的定義與三和弦

和弦根據牛津字典上的定義，是一群（通常是三個或以上）的音符因為和諧的緣故，齊奏所組成。這些音符之間的頻率是有一定規則的，其中「半音」的差距便是兩個音符的頻率相差  $\sqrt[12]{2} \approx 1.05946309$  倍，而一般所謂調號相同、卻差了「八度」的音，例如中音 A 與高音 A，實際上就是隔了十二個半音的距離，也就是頻率相差兩倍。

在一般的大三和弦中，是以一個音為「根音」之後，配合兩個分別比它高 4 與 7 個半音的音，即可形成一個大三和弦。小三和弦與大三和弦是幾乎一樣的，然而第二個音與根音之間僅有間隔 3 個半音。

而和弦也是可以「轉位」的，也就是說，和弦內的這些音只要音名與升降記號是正確的，可以自由的升或降不限制個數的八度。如此一來，同一個和弦的音可以同時被多種樂器所呈現，進而製造出和諧的背景音樂。

##### 2. 調外音與其對和弦配置的可能影響

在配上和弦的時候，比較保守的作法是盡量使多一點主旋律的音是和弦上的音。而並不是和弦內的音，一般又稱作「調外音」。然而除了考慮調外音是否會過多之外，配置一段主旋律使用的和弦往往還需要考量在其之前的主旋律或在其之後的主旋律以及和弦的選擇來做決定。然而這些都只是比較基本的想法，卻沒有固定的規則可循。

事實上，網路上有超過 150 支樂理教學影片的作曲家與鋼琴教師——官大為曾經創作過一首從頭到尾的主旋律都只有一個音，卻配上了許許多多不同和弦的實驗性樂曲。這也大致說明了任何的音跟任何的和弦其實都有產生和諧組合的潛力存在，也因此我在這個研究中會以市面上常見的熱門流行歌為主要對象。

### 3. 雷曼理論、和弦的作用與它們的稱呼

不同的和弦分別有順序的出現，稱為一組「和弦進行」。在一個和弦進行裡面，通常會有三種分類：主和弦（Tonic Chord）、下屬和弦（Subdominant Chord）與屬和弦（Dominant Chord）。

主和弦是以整首歌的調音為根音，也因此通常拿來開始或結束，也是一首歌裡面最穩定的狀態。屬和弦則恰恰相反，是發展到比較不確定、高潮的部分，通常後面會接上主和弦作收。下屬和弦則是具有從主和弦發展到屬和弦的過渡性質。以 C 大調為例的話，C 大三和弦就是主和弦、G 大三和弦就是屬和弦、F 大三和弦就是下屬和弦。

事實上，其他的和弦只要跟這三種和弦有重複的音，重複的部分越多，性質就會越相似。例如同樣以大調為例的話，E 小三和弦在 C 大調當中就會有介於 C 大三和弦與 G 大三和弦，也就是介於主和弦與屬和弦中間的性質。這樣的原因就在於它的三個音中，與另外兩個和弦都各有兩個音的重複。

#### （二） Music XML

Music XML 是 XML 的一種，是一種標記式語言，而以下就以英國歌手紅髮艾德（Ed Sheeran）所做的「高威女孩（Galway Girl）」的一段來做為說明。

```
69 | <part id="P1">
70 |   <measure number="1" width="393.60">
71 |     <print>
72 |   <attributes>
73 |   <note default-x="75.17" default-y="10.00">
74 |     <pitch>
75 |       <step>A</step>
76 |       <octave>5</octave>
77 |     </pitch>
78 |     <duration>2</duration>
79 |     <voice>1</voice>
80 |     <type>eighth</type>
81 |     <stem>down</stem>
82 |     <beam number="1">begin</beam>
83 |   </note>
```

圖 4-1 〈高威女孩〉的 Music XML 檔（節錄）

在從這種檔案裏面擷取資料的時候，比較會使用到的是以下列出的幾個標籤。

1. measure number (第幾個小節)
2. note (音符)
3. pitch/step (音高／音名)
4. duration (音符持續的時間，數值在不同的樂譜檔案中會受到連音與 $2^n$ 音符的數目影響，但每個小節的 duration 值是固定的)

### (三) 機器學習

#### 1. 中心思想

本次研究中所用到的規範式機器學習，其實和回歸問題是相似的，也就是說先收集大量的數據，包含了一些變數與希望從這些變數判斷的輸出值，有點類似「題庫」與「正確答案」，稱為「訓練集」或是「訓練資料」；然後利用一個函數，利用數學的參數優化方法，以期能夠建構一個模型，可以有效「解釋已知的數據關係，並成功預測接下來的輸入應有的輸出」，有如「練習附有正確答案的考古題以提升解新题目的技巧」一般。國立臺灣大學開發的 LIBSVM 機器學習工具裏的圖形介面，則可以用來說明「回歸」的概念是如何延伸到分類問題上的。在這個例子裏面，它以二維的參數構成不同的面，而要預測一個單一的資料點，便可以依據它所在的區塊來進行分類。



圖 4-2 LIBSVM 的圖形介面

隨著數學方法在機器學習的應用更廣、電腦運算也更加進步，現在的機器學習模型往往相當複雜，也因此會出現「訓練集的解釋能力優異，卻對預測新的結果表現差勁」的問題，這個現象通常被叫做「過適

(Overfitting)」。這就有點像是「考古題做的太熟以至於不知變通」；又或者像是以下的圖片一樣，藍色的線為了完全擬合訓練資料，有了不少的彎曲，然而對預測新的資料卻沒有任何準確度的幫助，反而可能使其下降。

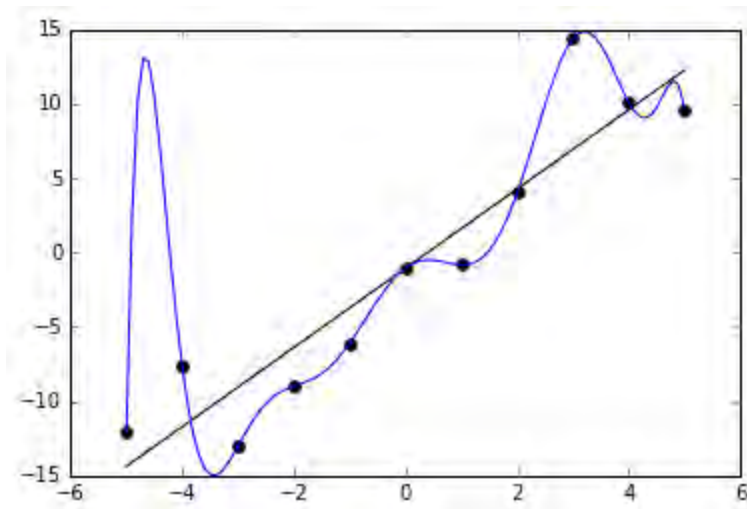


圖 4-3 過適現象的舉例

本研究中，便是將主旋律全部移調成 C 大調/A 小調音階，並把結果的分類分為 C (音名後面沒有符號為大三和弦，m 為小三和弦)、Dm、Em、F、G、Am 六類。(以 B 為根音且所有的音都在 C 大調音階上的和弦 Bdim 沒有 G 音的極為罕見，因此列入 G 計算)。

## 2. 隨機森林演算法

本研究使用隨機森林實作機器學習。隨機森林顧名思義是由許多的樹所組成。樹是一種資料結構，它長得像以下的圖：

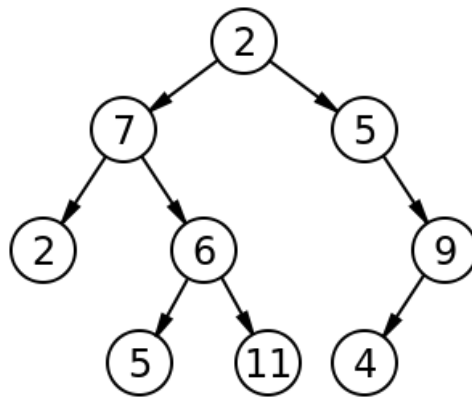


圖 4-4 一棵樹的示意圖

以上的每一個圓就是一個「節點」，而且這些節點由比它上面的節點延伸出去，因此它們有所謂的「分層」，而最大的層數就是所謂的「深度」。沒有向下層連結其他節點的節點又稱為「葉子」，而這個名稱也會在等等的參數調整中看到。

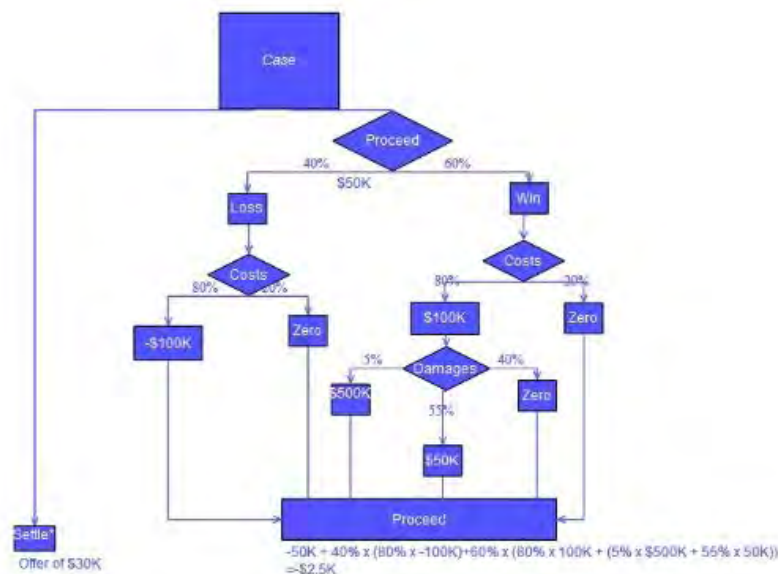


圖 4-5 決策樹的範例

決策樹則是一種樹，但是節點的內容除了葉節點皆由判斷式所構成，而每一個葉節點都是一種決策的結果，如上圖 4-5 所示。值得注意的是，樹由上往下的節點其實是不能收合、只能分岔的，因此在上圖中，最底下的 Proceed 步驟應該要有 6 個葉節點。

隨機森林中的個別決策樹便是將整個訓練集的資料隨機的取出（同一筆資料可能會被重複選到），並以這些資料建構決策樹模型。由於每一棵決策樹使用的資料不完全相同，每一棵建構出來的樹會進行「投票」；也就是



說，所有的樹都會產生一個預測結果，而由最多棵樹預測的結果為最終結果。

## 二、資料收集

- (一) 先從 YouTube 上選取它的推薦曲目，並透過它的自動撥放功能尋找作為訓練集的素材。這保證了歌曲的暢銷性，因此訓練資料多屬於市面上常見的流行歌，而非單一樂風的歌曲所組成。再從 MuseScore.com 尋找與原曲的主旋律相同的樂譜，並以 MuseScore File 的檔案格式下載。或者是利用亂數隨機選取 Wikifonia 中的 Music XML 檔案。
- (二) 接著把檔案內的內容使用 MuseScore 檔案進行編輯，轉成 C 大調或是 A 小調，並去掉重複的段落、修正撰譜者的錯誤等。編輯好之後，使用 MuseScore 內建的功能將其以 XML 檔案輸出。
- (三) 在 Spyder 的編輯環境下，我利用元素樹 (Element Tree) 的函式庫自己撰寫 Music XML 的解析程式碼，最後會得到數個九維的向量，每一個向量表示每半個小節的資料，概念如下所示：

表 4-1 我使用的訓練集的輸入之向量示意圖

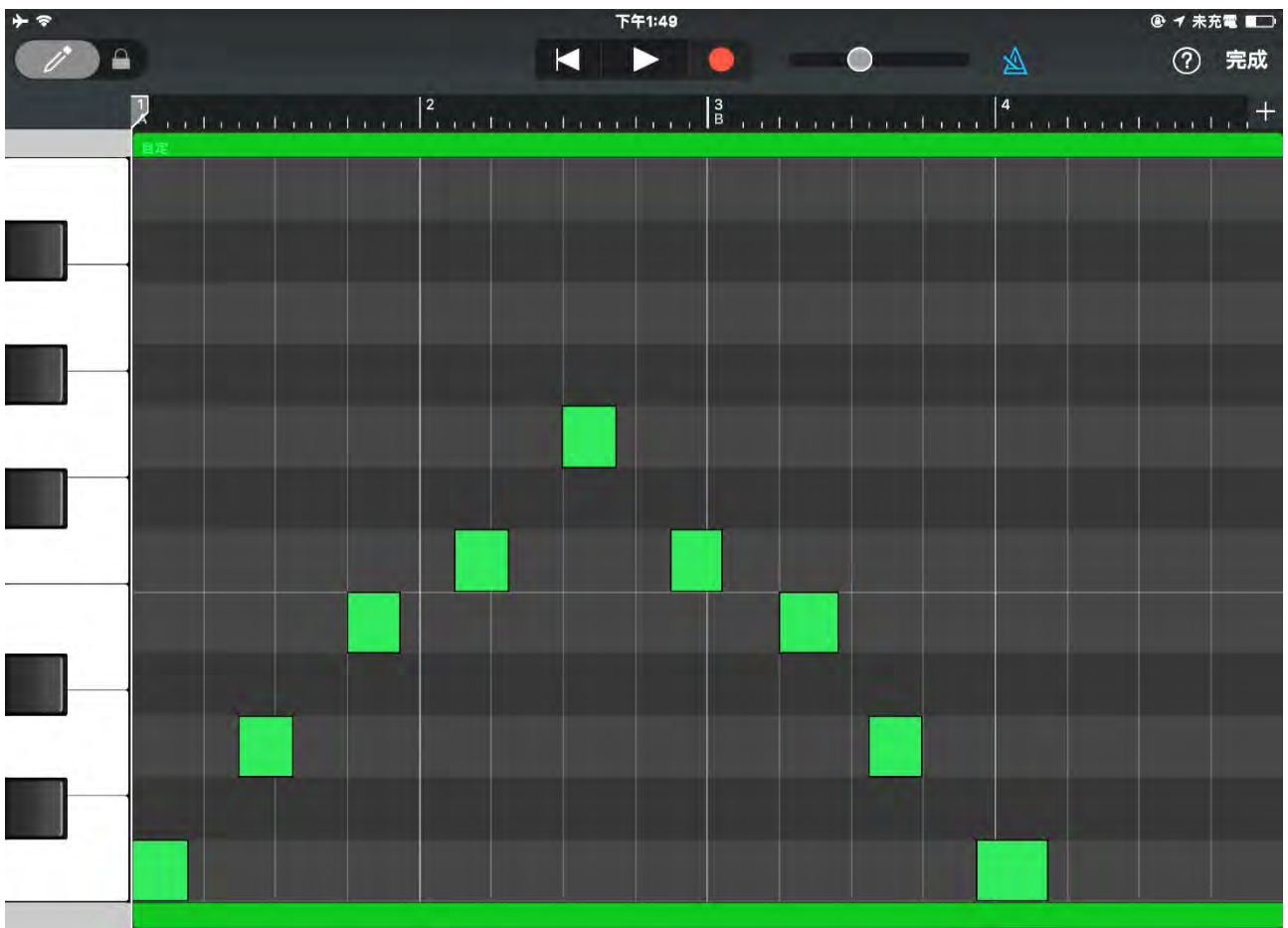
第 1 個 十六分 音符	第 2 個 十六分 音符	第 3 個 十六分 音符	第 4 個 十六分 音符	第 5 個 十六分 音符	第 6 個 十六分 音符	第 7 個 十六分 音符	第 8 個 十六分 音符	循環 代碼
--------------------	--------------------	--------------------	--------------------	--------------------	--------------------	--------------------	--------------------	----------

其中前八項的範圍在 1~12 之間，以該音的音名加上升降之後的值為結果，而位在哪一個八度上則不影響。換另一種方式來說，就是從頻率低於 (或等於) 該音的 C 音為 1 開始數，往上高了一個半音，值就會加一。例如說 B $\flat$ 與 A $\sharp$ 在該向量內，由於都是比 C 要高出 10 個半音，因此值都是  $1 + 10 = 11$ 。然而高音 A $\sharp$ 與中音 A $\sharp$ 的數值則是相同的，因為它們都是由比它們低的最接近 C 音數起。

其中就算因為有三連音等的關係，也會在程式的設計中，因為會將音符的長度作「量化 (Quantization)」而被概化成十六分音符的組合。下圖便是將量化的概念圖像化的示意圖。



圖 4-6 量化前（上，圖 4-6-1）與後（下，圖 4-6-2）的 MIDI 訊號



雖然視覺上要發覺兩者最不同之處可能會有困難，不過如果使用最上面的一排、以十六分音符為單位的刻度去對齊綠色的音符的話，圖 4-6-2 中所有的音符都會對在刻度上，而圖 4-6-1 中則否。這便是量化大概的效果：它會把所有的音符，不管實際時間長短為如何，都處理成整數向量的格式。雖然會有些許的失真，不過解析度達到一定水準其實就不會造成太大的影響。而機器學習裡面，對於輸入的準則通常就是以「真人可以做出判斷所需要用到的依據」為準，而已我的個人經驗而言，量化造成的失真小到幾乎可以忽略。

至於「循環代碼」則是表示該「半小節」的資料在一首歌的主歌或是副歌裡面常常出現的四小節循環或是八小節循環。對於配和弦來說，一個小段落在一首主歌或是副歌的位置，也是一個重要的決定因素。目前由於判斷的難度有點高，因此仍然以人為輸入為主。

關於訓練集中的「正確結果」，是透過我自己先在 YouTube 上聽得到的結果（市面上的譜往往有一些錯誤），並以根音分為六種和弦。舉例來說，有一首歌可能出現了 D 和弦，但是由於根音仍然是 D 音，因此列入 Dmin 計算。

### 三、使用 Python 與 scikit-learn 進行訓練、評分

在使用 scikit-learn 的時候，可以針對隨機森林有一些可調整的參數做調整。另外，由於樂理上通常假設和弦擁有越多相同的音，相似度越高，因此重複的音如果有  $n$  ( $0 \leq n \leq 3$  且  $n \in \mathbb{N}$ ) 個就算是  $\frac{n}{3}$  分。以下研究結果的圖表所顯示的數據，有些就是利用這個計算方式得到的結果之平均值（以下簡稱 W 值）。由於過程隨機，以下的 W 值都是訓練十次取的平均 W 值與標準差。

### 四、探討循環代碼對模型預測之影響

在本研究中，循環代碼是唯一可以標誌一筆訓練資料在一個樂句中的相對位置的特徵值。為了證實循環代碼對於模型預測準確度的影響，本研究中比較了保留、移除、與隨機給定循環代碼三種方式的模型預測準確度，並以一筆資料中的第一個十六分音符為對照組檢驗。由於一個樂句的結尾往往會使用一級和弦收尾或是五級和弦以連接下一個樂句，本研究中也特別討論循環代碼為 16 的資料。

## 伍、研究結果

以下所使用的訓練資料與測試資料為隨機分配，總共由 21 首歌中，1340 筆資料（也就是半小節）所組成。其中有 840 筆為訓練用，500 筆為測試用。以下所有的 W 值都是經過十次隨機的交叉比對；換言之，每一次的訓練資料都不盡相同。標準差則是套用 Microsoft Excel 的 STDEV.P 函數計算的母體標準差值。

### 一、循環代碼與 W 值的關係

表 5-1 對訓練資料之循環代碼採取之動作對 W 值之影響

對一筆資料採取之動作	W 值	標準差 ( $10^{-2}$ )
無	0.645	1.61
移除循環代碼	0.606	1.72
將循環代碼以亂數取代	0.563	1.39
移除第一個十六分音符	0.632	1.92
將第一個十六分音符以亂數取代	0.603	1.33

### 二、循環代碼為 16 之資料預測之和弦分布

以下表格中的數字是模型預測結果與原始結果相吻合的資料筆數，而傳統正確率則是上述的資料筆數加總除以總資料筆數得到的比值。

表 5-2 對訓練資料之循環代碼採取之動作對和弦預測之準確度分布情形之影響

對一筆資料採取之動作	C	Em	G	Dm	F	Am	傳統正確率
原曲	18	11	33	7	7	7	1.000
無	11	7	27	5	4	2	0.675
移除循環代碼	10	5	21	4	3	2	0.542
將循環代碼以亂數取代	9	3	14	1	3	2	0.386

## 陸、討論

### 一、模型的預測結果分布分析

#### (一) 混淆矩陣 (Confusion Matrix)

混淆矩陣就是為了比較「正確答案」與「實際預測結果」的差異，而和弦因為相同的音形成的「相似度」理論上可以用以下「三度環 (Cycle of Thirds)」進行解釋。三度環可以保證相鄰的和弦各有兩個相同的音，也往往互相取代性較高。

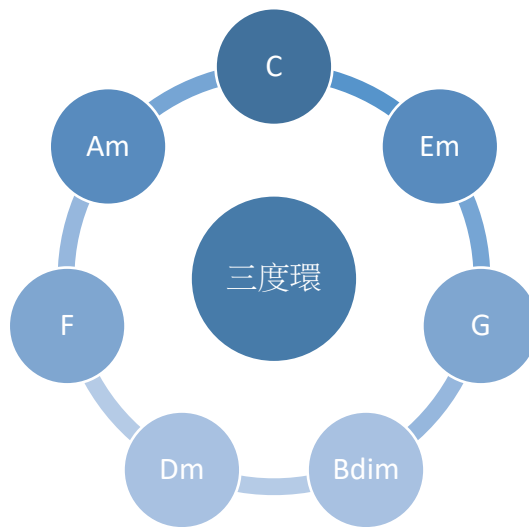


圖 6-1 三度環

表 6-1 混淆矩陣

預測結果 標準答案	C	Em	G	Dm	F	Am	總和
C	54.9	3.6	5.7	8.7	19.4	11.5	103.8
Em	6.7	7.7	10.5	4.6	9.3	5.7	44.5
G	5.6	4.5	38.3	7.0	14.7	5.0	75.1
Dm	10.7	1.3	7.6	24.6	14.4	8.8	67.4
F	15.4	5.0	16.3	8.9	66.3	13.2	125.1
Am	10.0	2.5	5.8	9.6	15.3	40.9	84.1
總和	103.3	24.6	84.2	63.4	139.4	85.1	500.0

上表中有「預測結果為 X 且結果為 Y」的資料筆數 Z，而在下表中，Z 會被除以「以 X 為標準答案的資料筆數 × 以 Y 為標準答案的資料筆數」。這是因為當一個和弦出現的頻率較高時，被預測的總數就會隨之提高，因此調整出「機率」以完整分析便是這個操作的重點所在。下表的數據單位為 $10^{-3}$ 。

表 6-2 經調整後的混淆矩陣

預測結果 標準答案	C	Em	G	Dm	F	Am
C	5.10	0.78	0.73	1.24	1.49	1.32
Em	1.45	3.89	3.14	1.53	1.67	1.52
G	0.72	1.35	6.79	1.38	1.56	0.79
Dm	1.53	0.43	1.50	5.42	1.71	1.55
F	1.19	0.90	1.73	1.06	4.24	1.25
Am	1.15	0.67	0.92	1.69	1.45	5.78

(二) 機率陣列的加總

如果以 $P_X$ 表示 X 和弦為該筆資料的答案的機率，則該陣列長相應會是

$$[P_C \ P_{Em} \ P_G \ P_{Dm} \ P_F \ P_{Am}]$$

下表與上表不同的是這次數據是以機率的方式相加。

表 6-3 機率加總的混淆矩陣（以下簡稱機率矩陣）

預測結果 標準答案	C	Em	G	Dm	F	Am	總和
C	39.8	7.2	8.9	10.6	18.3	13.9	98.7
Em	7.9	8.1	9.2	5.2	9.2	6.8	46.3
G	8.8	8.5	26.5	8.2	14.9	8.5	75.5
Dm	11.4	4.5	8.0	18.9	14.0	9.5	66.3
F	20.8	9.0	17.1	15.3	48.5	16.1	126.8
Am	13.6	5.9	8.6	9.7	16.3	29.3	83.3
總和	102.3	43.2	78.3	67.9	121.1	84.1	496.9

下表的數據單位為 $10^{-3}$ 。

表 6-4 經調整後的機率矩陣

預測結果 標準答案	C	Em	G	Dm	F	Am
C	4.09	1.57	1.19	1.62	1.46	1.70
Em	1.72	3.78	2.63	1.69	1.56	1.76
G	1.18	2.44	4.65	1.63	1.56	1.35
Dm	1.74	1.46	1.60	4.31	1.66	1.72
F	1.66	1.54	1.79	1.82	3.02	1.52
Am	1.65	1.52	1.36	1.76	1.54	4.22

(三) 不同和弦之間「伴隨」的可能性

下表同樣有利用到機率陣列，而表格中的結果就是將每一筆的資料中，機率最高的兩個預測結果整理出來。往往會一起成為判斷結果，也可能代表相似度。

表 6-5 和弦的關聯矩陣（以下簡稱關聯矩陣）

預測結果 標準答案	C	Em	G	Dm	F	Am
C	0.00	3.35	1.93	2.31	2.65	3.42
Em	0.90	0.00	2.26	0.75	1.29	1.32
G	1.80	3.98	0.00	2.72	3.18	1.56
Dm	1.62	1.34	2.18	0.00	2.81	1.96
F	2.77	1.94	3.41	3.25	0.00	3.00
Am	3.04	3.53	1.77	1.79	2.43	0.00

由於上表的數據只有最大與次大的關係，因此下表使用的修正使用表 6-5 的總和進行修正。下表的數據單位為 $10^{-3}$ 。

表 6-6 經調整後的機率加總混淆矩陣

次大值 最大值	C	Em	G	Dm	F	Am	總和
C	0.0	15.3	14.4	15.1	33.2	28.1	106.1
Em	4.1	0.0	7.9	2.3	7.6	5.1	27.0
G	13.4	13.9	0.0	13.6	30.5	9.8	81.2
Dm	10.6	4.1	10.9	0.0	23.6	10.8	60.0
F	34.7	11.4	32.6	27.3	0.0	31.7	140.2
Am	22.6	13.6	11.1	9.9	25.7	0.0	90.7
總和	85.4	58.3	76.9	68.2	123.1	93.3	505.2

## 二、混淆矩陣的分布討論

使用表 6-2 以及表 6-4 可以整理出最容易與最不容易混淆的和弦組合。

- (一) C 和弦除了特別不容易和 G 和弦搞混。
- (二) Em 和弦最容易跟 G 和弦搞混。
- (三) G 和弦特別不容易和 C 和弦與 Am 和弦搞混，但是容易與 Em 和弦搞混。
- (四) Dm 和弦不容易與 Em 和弦搞混。
- (五) F 和弦不容易與 Em 和弦搞混。
- (六) Am 和弦不容易與 G 和弦與 Em 和弦搞混。

## 三、和弦的伴隨現象討論

- (一) 判斷為 C 和弦的資料第二選擇往往是 Em 或 Am 和弦，G 和弦最少。
- (二) 判斷為 Em 和弦的資料第二選擇往往是 G 和弦，C 或 Dm 和弦則最少。
- (三) 判斷為 G 和弦的資料第二選擇往往是 Em 或 F 和弦，Am 和弦則最少。
- (四) 判斷為 Dm 和弦的資料第二選擇往往是 F 和弦，Em 和弦則最少。



(五) 判斷為 F 和弦的資料第二選擇往往是 G 或 Dm 和弦，Em 和弦則最少。

(六) 判斷為 Am 和弦的資料第二選擇常是 Em 或 C 和弦，G 或 Dm 和弦最少。

#### 四、循環代碼的作用

如表 5-1 與 5-2 所示，循環代碼在移除時，造成的 W 值下降較對照組：第一個十六分音符要多，因此代表循環代碼對於模型預測的準確度比其他的特徵值相對而言要重要。另外在表 5-2 中也可以看出循環代碼為 16 的資料（也就是樂句的結尾）大部分皆為配上 C 或是 G 和弦，而這種分布的不均也是在稍早的混淆矩陣討論中必須要做調整的原因。而從表 5-1 與 5-2 中的數據都可以得到同樣的模式，那就是一個參數被亂數取代比起剔除掉該參數會對準確率有著更大的負面影響。

### 柒、結論

一、和弦間容易混淆的程度與和弦在三度環上的位置接近或是遠離有著顯著的關係。

二、樂理中的主和弦與屬和弦的分別在數據上特別明顯，代表這個模型證實了兩種和弦樂理上存在的差別（在樂理中，主和弦最為穩定，而屬和弦最具緊張感）。

三、循環代碼的使用可以大幅提高和弦預測的準確度，代表旋律在一段樂句中的位置對於和弦配置的重要性不亞於和旋律的組成音本身。

四、之後可以嘗試將這項技術應用到音樂創作，例如樂譜編輯軟體的外掛程式開發。

### 捌、參考資料、圖片來源及附錄

#### 一、參考資料

(一) Download for Wikifonia all 6,675 Lead Sheets! | General Arranger Keyboard Forum | Synth Zone Forums. General Arranger Keyboard Forum | Synth Zone Forums. Retrieved February 18, 2018, from

[http://www.synthzone.com/forum/ubbthreads.php/topics/384909/Download\\_for\\_Wikifonia\\_all\\_6\\_6](http://www.synthzone.com/forum/ubbthreads.php/topics/384909/Download_for_Wikifonia_all_6_6)

(二) chord | Definition of chord in English by Oxford Dictionaries. Oxford University Press. Retrieved February 17, 2018, from

<https://en.oxforddictionaries.com/definition/chord>

- (三) Harmonic functions. In K. Shaffer, B. Hughes, B. Moseley, Open Music Theory. Hybrid Pedagogy Publishing. Retrieved February 18, 2018, from <http://openmusictheory.com/harmonicFunctions.html>
- (四) 官大為(2016年2月20日)•旋律是什麼音的時候可以配上 C7 和弦?(提示:比你想像中還多) - YouTube • 取自 <https://www.youtube.com/watch?v=MgT-skV8QGA>
- (五) Chih-Chung Chang, Chich-Jen Lin (2011). LIBSVM: A library for support vector machines [Abstract]. ACM Transactions on Intelligent Systems and Technology (TIST), Volume 2 Issue 3, April 2011, Article No.27. Retrieved February 18, 2018, from <https://dl.acm.org/citation.cfm?id=1961199>

## 二、圖片來源

- (一) Overfitting - Wikipedia. Wikipedia. Retrieved February 18, 2018, from <https://en.wikipedia.org/wiki/Overfitting>
- (二) Tree (data structure) - Wikipedia. Wikipedia. Retrieved February 18, 2018, from [https://en.wikipedia.org/wiki/Tree\\_\(data\\_structure\)](https://en.wikipedia.org/wiki/Tree_(data_structure))
- (三) Decision tree - Wikipedia. Wikipedia. Retrieved February 18, 2018, from [https://en.wikipedia.org/wiki/Decision\\_tree](https://en.wikipedia.org/wiki/Decision_tree)

## 三、模型預測使用之歌曲

- (一) 盧廣仲 - 魚仔
- (二) 李千娜 - 不曾回來過
- (三) 鄧子祺 - 光年之外
- (四) 黃明志 - 漂向北方
- (五) 聖結石 - 真的不想嘴
- (六) 聖結石 - 安儷
- (七) Ed Sheeran - Galway Girl
- (八) Ed Sheeran - Shape of You
- (九) Clean Bandit - Rockabye

- (十) Bruno Mars - That's What I Like
- (十一) Bruno Mars - Versace on the Floor
- (十二) Bruno Mars - Lazy Song
- (十三) Bruno Mars - Treasure
- (十四) Bruno Mars - Locked out of Heaven
- (十五) Bruno Mars - Just the Way You Are
- (十六) Bruno Mars - Grenade
- (十七) Louis Tomlinson - Back to You
- (十八) Lil Wayne - Strip That Down

四、模型預測成果：以光良〈童話〉之副歌為例

## 童話

光良

Chords for the first staff: C, C, C, Am, F, F, Em, Am

Chords for the second staff: Am, F, Am, Am, Dm, G, C, C

五、實作 Music XML 解析的程式碼：以 Ed Sheeran 的〈Shape of You〉為例（見下頁）

```

import xml.etree.ElementTree as ET

SOY = ET.fromstring(data)
measures = SOY.findall('part/measure')

d10 = [[0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]] #Data Set

dphb = 4 #Duration per half bars
lp16 = 0 #8-bar loop counter, in the range between 1 ~ 16 after initialisation

for measure in measures:
    curr = 0 #Current place in a bar, in the range between 0 ~ (2 * dphb)
    Tset = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] #Note data of the measure
    notes = measure.findall('note') #Array of notes

    for note in notes:
        if not(note.find('pitch/step') is None):
            a = note.find('pitch/step').text #Text that describes the pitch
        else:
            a = 'L'
        p = 0 #Pitch data in integers, in the range between 0 ~ 12

        #Setting the value of p
        if(a == "C"):
            p = 1

        if(a == "D"):
            p = 3

        if(a == "E"):
            p = 5

        if(a == "F"):
            p = 6

        if(a == "G"):
            p = 8

        if(a == "A"):
            p = 10

        if(a == "B"):
            p = 12

        if not(note.find('pitch/alter') is None):
            p += int(note.find('pitch/alter').text) #Adding alters to the note

        d = int(note.find('duration').text) #Duration of the note, in the range between 0 ~ (2 * dphb)

        alpha = int(round(curr / dphb * 8)) #The 16th-note-wise starting point of the note, in the range between 0 ~ 15
        beta = int(round((curr + d) / dphb * 8)) #The 16th-note-wise ending point of the note, in the range between 0 ~ 15

        for i in range(alpha, beta):
            Tset[i] = p #Assigning the pitch to measure's note data

        curr += d #Moving current point by duration

        if(curr >= 2 * dphb): #If current point reaches the end of the measure
            #Loop counting and mod by 16
            lp16 += 1

            if lp16 > 16:
                lp16 -= 16

            #Splitting into half-bar data
            hsss = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
            hsss[8] = lp16 #Assigning 8-bar loop value

            #Assigning the first half of Tset
            for i in range(0, 8):
                hsss[i] = Tset[i]

            d10.append(hsss)

            lp16 += 1

            if lp16 > 16:
                lp16 -= 16

            hsss = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
            hsss[8] = lp16

            for i in range(8, 16):
                hsss[i - 8] = Tset[i]

            d10.append(hsss)
            break

del(d10[0])
del(d10[0])

```

## 【評語】 052503

本研究探討如何以機器學習的角度，讓機器學習樂理中和弦功能與歌曲旋律的相互關係。配和弦之具體化創作構想，源自於學生對音樂的興趣與專長，研究成果亦富有獨特研究價值。

研究過程的描述詳盡，對於使用到的技術也有一定程度的理解，不管是提升學生的研究能力或是研究主題的貢獻皆有不錯的提升。其研究成果與結論對未來音樂創作等應用，帶來不錯的建議。

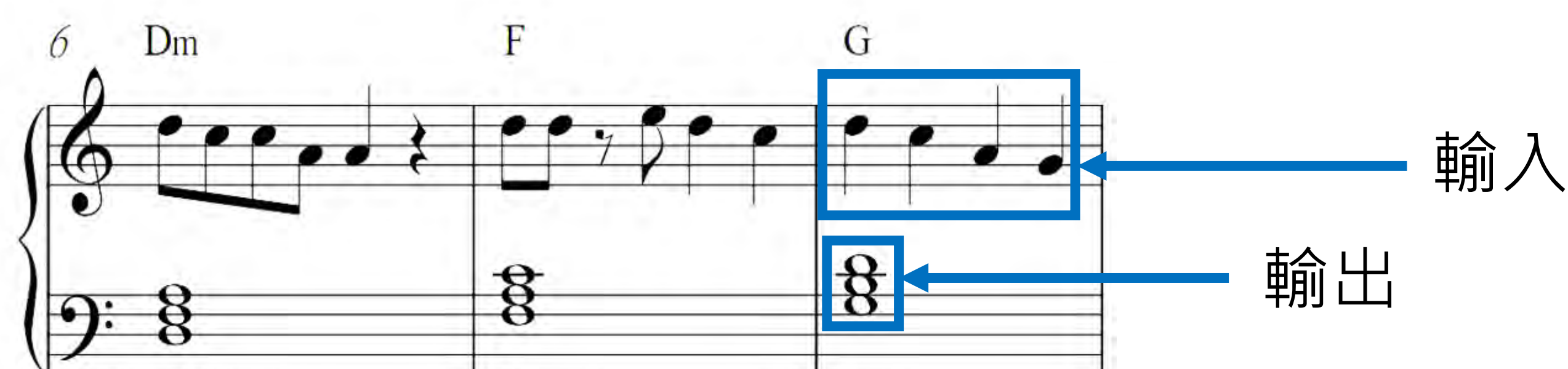
# 研究概述

本研究使用了隨機森林模型實作對主旋律的和弦配置，並同時透過預測結果佐證了樂理上不同的和弦之相關性，與一個和弦在一段和弦進行中較常扮演的角色。在實作的過程中，透過輸入與輸出的內容與格式改變，可以證實一段**旋律在樂段中的位置對於預測的準度相當的重要**。

# 研究目的

- 建構與訓練機器學習模型，以主旋律為唯一的輸入，輸出該段主旋律的和弦進行。
- 分析模型預測時易與正確答案混淆的和弦分布情形，佐證樂理中和弦「相似度」的關係與和弦的功能，**使樂理中情感化的敘述得到科學解釋**
- 透過輸入的資料差異，比較大小調、one-hot輸入、循環代碼的有無、音符的短少等因素對於模型預測準確度的影響

# 樂理名詞解釋

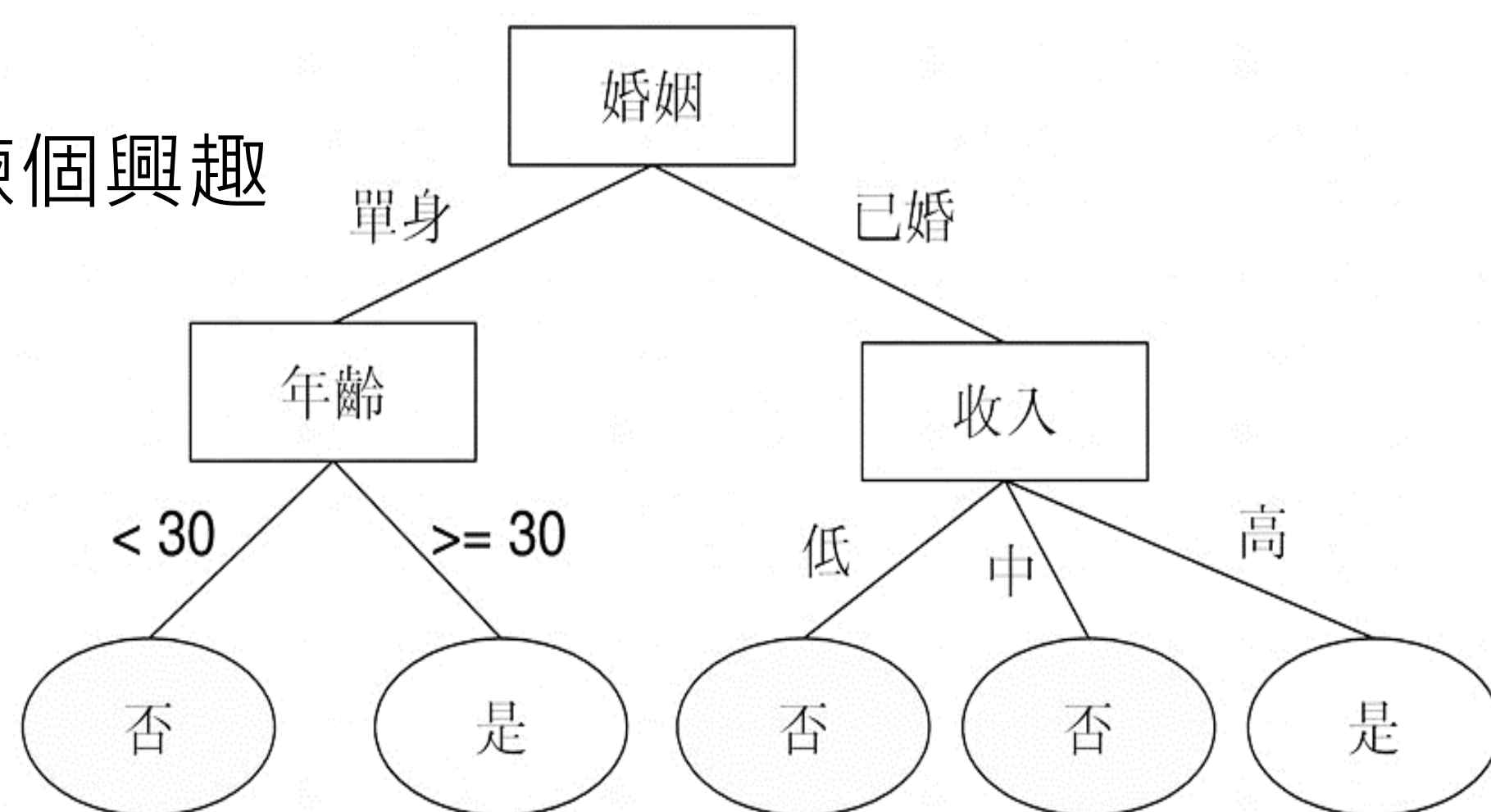


1. 和弦：一群（通常是三個或以上）的音符因為和諧的緣故，齊奏所組成。
2. 度數：一首歌的大調音階中的七個音從該調的根音（如E大調中的E音）開始數，就分別為一到七度，在紀錄和弦記號時以羅馬數字表示。
3. 三和弦：以一個音為和弦的根音，以它與它的三度與五度音組合而成的和弦，如以C為根音，C、E、G三個音所構成的C和弦。
4. 大、小和弦：三和弦中的第二個音比第一個音高出四個半音（高一個半音頻率是 $12\sqrt{2} \approx 1.05946309$ 倍）叫做大三和弦，如果是三個半音就是小三和弦。
5. 主、屬、下屬和弦：主和弦（I和弦）最穩定，用於開始與結束居多；屬和弦是較為緊張的，之後往往會直接接回主和弦；下屬和弦則可以做為主和弦到屬和弦的過渡。

# 應用的電腦技術

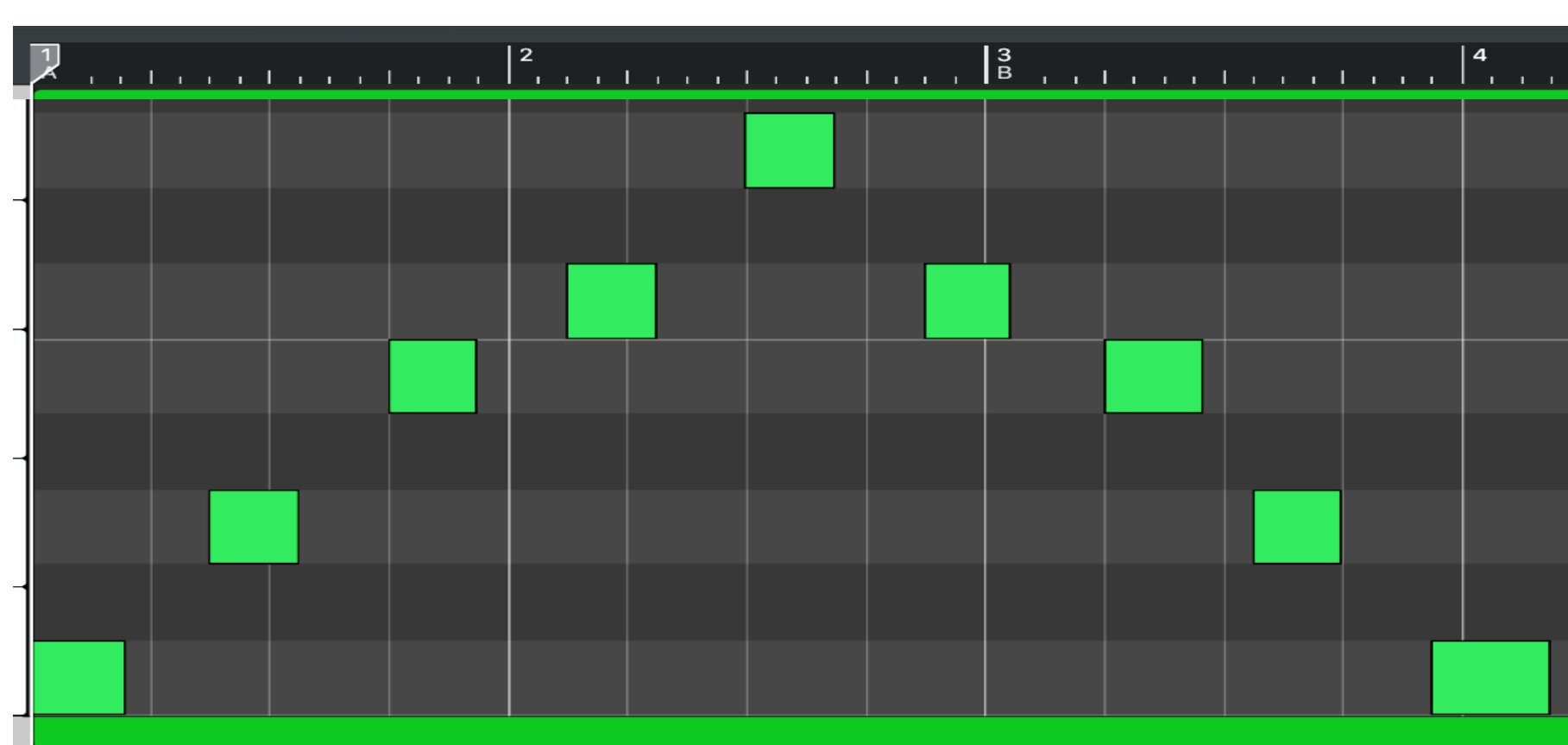
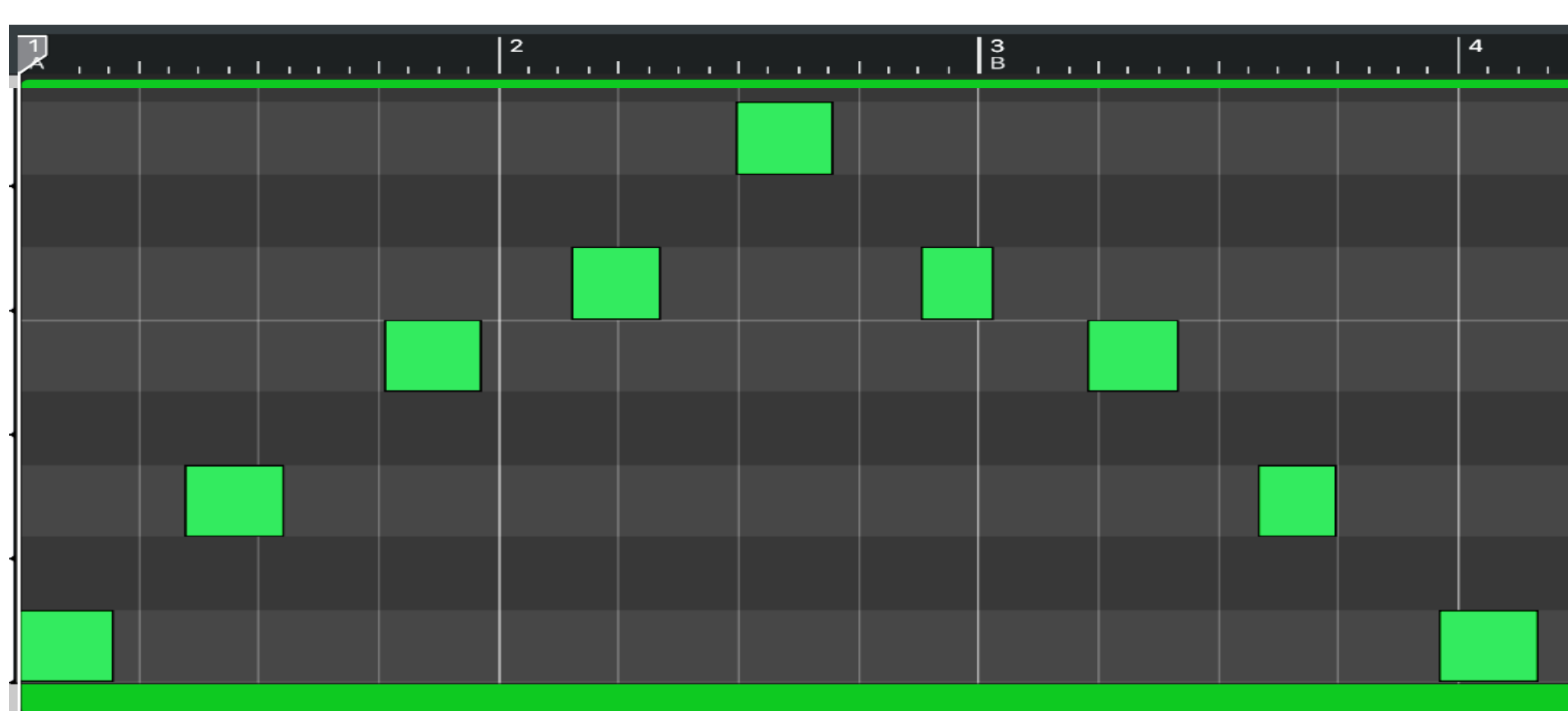
例：要不要買樂器練個興趣

- Music XML：XML的一種，一種標記式語言
- 機器學習：「解釋已知的數據關係，並用它成功預測接下來的輸入應有的輸出」



- 隨機森林：許多決策樹投票決定結果，舉例如右
- **量化**：將各種不同的連音概化成十六分音符的組合

下為量化前與量化後的比較圖，左邊沒有對齊刻度，右邊有



決策樹的特點

1. 有明確不可擾亂的分層
2. 只有分岔，沒有收合
3. 最末端（葉節點）是判斷結果
4. 模型解釋能力強

# 資料收集：

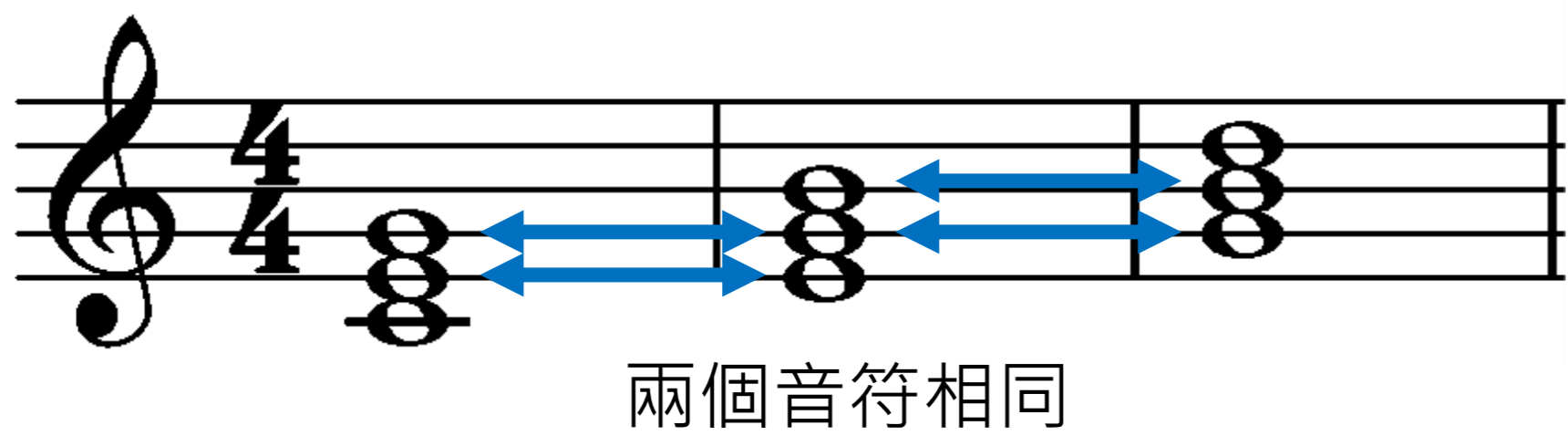
22首歌，共1340筆

- 尋找歌曲**
  - YouTube的自動推薦播放歌單
  - 以**流行歌**為資料，不限音樂風格
- 尋找對應樂譜**
  - MuseScore.com上的對應樂譜
  - 轉為C大調/A小調音階
  - 和弦由於常有錯誤，為手動輸入
- 將樂譜轉為Python陣列**
  - 把旋律**量化**並轉成下表的九維向量
  - 資料以半個小節為單位
  - **循環代碼**是該半小節在八個小節的循環中的位置

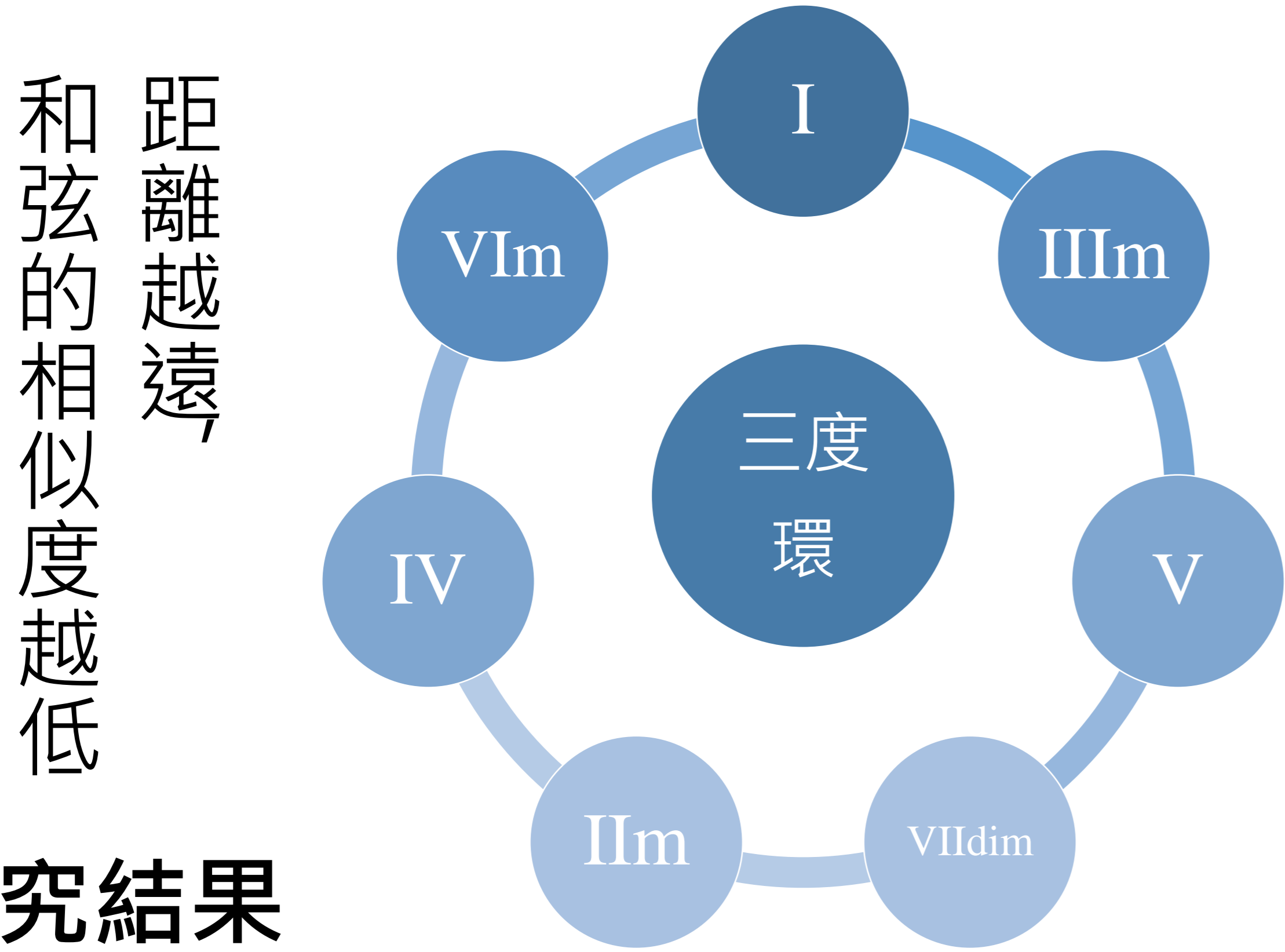
例：Ed Sheeran “Thinking Out Loud” 第二筆資料

	一般資料	One-hot輸入											
		C	C#	D	Eb	E	F	F#	G	G#	A	Bb	B
第1個音符	10	0	0	0	0	0	0	0	0	1	0	0	
第2個音符	10	0	0	0	0	0	0	0	0	1	0	0	
第3個音符	8	0	0	0	0	0	0	1	0	0	0	0	
第4個音符	5	0	0	0	0	1	0	0	0	0	0	0	
第5個音符	5	0	0	0	0	1	0	0	0	0	0	0	
第6個音符	3	0	0	1	0	0	0	0	0	0	0	0	
第7個音符	3	0	0	1	0	0	0	0	0	0	0	0	
第8個音符	1	1	0	0	0	0	0	0	0	0	0	0	
循環代碼	2	2											

# 模型訓練與評分



- scikit-learn：一個機器學習的Python函式庫，可以用以實驗不同的參數
  - 越多與正確的和弦相同的音，代表越精準（如上圖）
- 獨創的「三度環」與W值（和弦的音相同的比例）



# 研究結果

表一：循環代碼之有無對正確度影響

特徵值之移除與改變	平均W值	平均傳統正確率
有循環代碼與第一個十六分音符	64.6%	49.4%
僅去除循環代碼	60.4%	43.4%
僅將循環代碼以1~16的亂數取代	56.4%	37.2%
僅去除第一個十六分音符	63.5%	48.0%
僅將第一個十六分音符以1~12的亂數取代	59.5%	42.3%

表二：one-hot的使用對正確度影響

輸入資料之維度	平均W值	平均傳統正確率
九維陣列	63.3%	48.3%
將每個音符皆轉為十二維，共97維	65.8%	51.0%

表三：單位資料長短對正確度影響

一筆資料之長度	單位音符長	平均W值	平均傳統正確率
四分之一小節	三十二分音符	60.7%	43.4%
半小節	十六分音符	65.5%	50.9%
一小節	八分音符	70.8%	56.8%

表四：循環代碼對樂句末小節預測影響

對一筆資料採取之動作	預測結果						傳統正確率
	I	IIIIm	V	IIIm	IV	VIIm	
原曲	18	11	33	7	7	7	100%
無	11	7	27	5	4	2	67.5%
移除循環代碼	10	5	21	4	3	2	54.2%
將循環代碼以亂數取代	9	3	14	1	3	2	38.6%

# 預測結果分布討論

表五：隨機森林模型的混淆矩陣

預測結果 / 標準答案	I	IIIIm	V	IIIm	IV	VIIm	總和
I	54.9	3.6	5.7	8.7	19.4	11.5	103.8
IIIIm	6.7	7.7	10.5	4.6	9.3	5.7	44.5
V	5.6	4.5	38.3	7.0	14.7	5.0	75.1
IIIm	10.7	1.3	7.6	24.6	14.4	8.8	67.4
IV	15.4	5.0	16.3	8.9	66.3	13.2	125.1
VIIm	10.0	2.5	5.8	9.6	15.3	40.9	84.1
總和	103.3	24.6	84.2	63.4	139.4	85.1	500.0

單位：筆

表六：表一的數據經調整後之結果

預測結果 / 標準答案	I	IIIIm	V	IIIm	IV	VIIm
I	5.10	0.78	0.73	1.24	1.49	1.32
IIIIm	1.45	3.89	3.14	1.53	1.67	1.52
V	0.72	1.35	6.79	1.38	1.56	0.79
IIIm	1.53	0.43	1.50	5.42	1.71	1.55
IV	1.19	0.90	1.73	1.06	4.24	1.25
VIIm	1.15	0.67	0.92	1.69	1.45	5.78

單位：10<sup>-3</sup> 筆<sup>-1</sup>

「調整」是將數值除以兩個和弦的真實出現數目，如表二中C-Em的 0.78 × 10<sup>-3</sup>就是表一中3.6除以「總和」欄中的103.8與44.5的值

表七：調整後的類神經網路混淆矩陣

預測結果 / 標準答案	I	IIIIm	V	IIIm	IV	VIIm
I	3.82	0.02	1.20	0.07	3.41	1.06
IIIIm	1.42	0.21	2.14	0.07	4.24	1.72
V	1.38	0.05	2.95	0.17	4.09	1.24
IIIm	2.75	0.00	2.17	0.25	3.23	1.58
IV	2.76	0.05	1.73	0.13	3.75	1.27
VIIm	2.38	0.11	1.33	0.29	3.31	2.58

單位：10<sup>-3</sup> 筆<sup>-1</sup>

表八：當預測結果為機率時，機率最大與機率次大的關係

機率最大 / 機率次大	I	IIIIm	V	IIIm	IV	VIIm
I	0.00	3.35	1.93	2.31	2.65	3.42
IIIIm	0.90	0.00	2.26	0.75	1.29	1.32
V	1.80	3.98	0.00	2.72	3.18	1.56
IIIm	1.62	1.34	2.18	0.00	2.81	1.96
IV	2.77	1.94	3.41	3.25	0.00	3.00
VIIm	3.04	3.53	1.77	1.79	2.43	0.00

單位：筆

# 和弦相似度之討論

和弦	最相似	差別最大
I	III <sub>m</sub> 、VI <sub>m</sub>	V
III <sub>m</sub>	V	I、II <sub>m</sub>
V	III <sub>m</sub> 、IV	I、VI <sub>m</sub>
II <sub>m</sub>	IV	III <sub>m</sub>
IV	V、II <sub>m</sub>	III <sub>m</sub>
VI <sub>m</sub>	III <sub>m</sub> 、I	V、II <sub>m</sub>

## 結論

1. 循環代碼被移除時，造成的W值下降較對照組（移除第一個十六分音符）要多  
→ 循環代碼對於模型預測的準確度比其他的特徵值相對而言要重要
2. 循環代碼為16的資料（也就是樂句的結尾）大部分皆為配上I或是V和弦，而這種**資料分布的不均**的現象也是在稍早的混淆矩陣討論中必須要做調整的原因。
3. 一個參數被亂數取代比起剔除掉該參數會更加降低預測的準確度。

## 結論

1. **三度環上的位置**與和弦相似度有著顯著的關係。
2. 證實了主和弦和屬弦的極端性（樂理中主和弦最為穩定，而屬和弦最具緊張感）。
3. **循環代碼**可以大幅提高和弦預測的準確度。
4. 之後可以嘗試將這項技術**應用到音樂創作**的領域，例如樂譜編輯軟體的外掛程式開發。

## 成果範例

[6, 6, 6, 6, 6, 6, 8, 8, 1]  
 [8, 8, 8, 8, 10, 10, 10, 10, 2]  
 [12, 12, 12, 12, 12, 12, 1, 1, 3]  
 [1, 1, 1, 1, 3, 3, 3, 3, 4]  
 [1, 1, 1, 1, 1, 1, 12, 12, 5]  
 [12, 12, 12, 12, 12, 12, 12, 12, 6]  
 [12, 12, 12, 12, 12, 12, 12, 12, 7]  
 [0, 0, 0, 0, 5, 5, 5, 5, 8]  
 [6, 6, 6, 6, 6, 6, 8, 8, 9]  
 [8, 8, 8, 8, 10, 10, 10, 10, 10]  
 [12, 12, 12, 12, 12, 12, 1, 1, 11]  
 [1, 1, 1, 1, 3, 3, 3, 3, 12]

預測結果 原版的和弦

6	6	1
1	1	1
6	7	0.67
5	6	0.67
7	2	0.33
2	4	0.33
3	3	1
5	4	0.67
6	5	0.67
1	7	0.67
3	3	0.67
5	2	0

預測結果的W值

和弦碼	對應和弦
1	I
2	III <sub>m</sub>
3	V
5	II <sub>m</sub>
6	IV
7	VI <sub>m</sub>

自己寫的 Music XML 解析程式

輸入

利用訓練好的模型預測每半個小節的最適和弦

輸出的數字代表的是和弦記號

依據輸出的和弦製作的基本款伴奏

## 參考資料

1. Download for Wikifonia all 6,675 Lead Sheets! | General Arranger Keyboard Forum | Synth Zone Forums. General Arranger Keyboard Forum | Synth Zone Forums. Retrieved February 18, 2018, from [http://www.synthzone.com/forum/ubbthreads.php/topics/384909/Download\\_for\\_Wikifonia\\_all\\_6\\_6](http://www.synthzone.com/forum/ubbthreads.php/topics/384909/Download_for_Wikifonia_all_6_6)
2. chord | Definition of chord in English by Oxford Dictionaries. Oxford University Press. Retrieved February 17, 2018, from <https://en.oxforddictionaries.com/definition/chord>
3. Harmonic functions. In K. Shaffer, B. Hughes, B. Moseley, Open Music Theory. Hybrid Pedagogy Publishing. Retrieved February 18, 2018, from <http://openmusictheory.com/harmonicFunctions.html>
4. 官大為 (2016年2月20日) • 旋律是什麼音的時候可以配上 C7 和弦? (提示: 比你想像中還多) - YouTube • 取自 <https://www.youtube.com/watch?v=MgT-skV8QGA>
5. Chih-Chung Chang, Chich-Jen Lin (2011). LIBSVM: A library for support vector machines [Abstract]. ACM Transactions on Intelligent Systems and Technology (TIST), Volume 2 Issue 3, April 2011, Article No.27. Retrieved February 18, 2018, from <https://dl.acm.org/citation.cfm?id=1961199>
6. Overfitting - Wikipedia. Wikipedia. Retrieved February 18, 2018, from <https://en.wikipedia.org/wiki/Overfitting>
7. Tree (data structure) - Wikipedia. Wikipedia. Retrieved February 18, 2018, from [https://en.wikipedia.org/wiki/Tree\\_\(data\\_structure\)](https://en.wikipedia.org/wiki/Tree_(data_structure))
8. Decision tree - Wikipedia. Wikipedia. Retrieved February 18, 2018, from [https://en.wikipedia.org/wiki/Decision\\_tree](https://en.wikipedia.org/wiki/Decision_tree)