4.

在上一節課我們提過了機器學習是實現人工智慧的一種手段，而現在很火的深度學習則是機器學習的一個分支，他們的關西如上面這個圖異樣。

這邊用兩個簡單的例子去說明機器學習和深度學習不同的點在哪邊，假如我們要用機器學習建立一個颱風預測系統，我們首先可以去瀏覽歷年來颱風的資料，從這些資料中得出一些模式，這些模式包含了能導致颱風的具體條件，像是溫度大於40度 濕度在70到100之間之類的，而溫度與濕度這些指標就是機器學習裡的特徵，這些都是人工設計好的，也就是說我們在做這個預測系統的時候，會先由專家通過分析知道哪些特徵是重要的，然後電腦再通過分析歷史數據中的這些特徵來找出相應的模式，因此不同的特徵組合會有不同的結果

而深度學習則不一樣，他是透過模擬人類神經元的方式去做計算，簡單的例子，如果人類要去區分下面的圖娜個是方的哪個是圓的，我們眼睛第一件事情是判斷這個圖有沒有四條邊，如果有的話再看他們是不是連在一起，是不適等長與互相垂直的，如果都符合我們就會判斷他為正方形，從上面這個問題可以看出來，我們會將一個複雜的抽象問題分辨形狀，分解成比較簡單比較不抽象的小問題判斷邊長角度之類的，深度學習基本就是在模擬這個過程

以人臉識別任務來說，raw data就是原始數據，這個機器無法去理解是甚麼，於是深度學習首先盡可能地找到與這個頭像相關的各種邊角，這些邊角就是底層特徵，接下來對這些底層特徵進行組合，就可以湊出鼻子眼睛耳朵等東西，再進行組合則可以組出各種人臉，這個時候就能去辨識是否為人頭了

5.

了解工作過程後這邊簡單總結一下兩者差別，深度學習好壞很大部分取決於資料量，這張圖上可以看出當資料量多的時候dl比較好，少的時候則是ml

然後是dl很吃gpu，因為它很多任務都是用矩陣運算

最後dl不需要人工給特徵而ml要

6.

接下來說明一下ml的工作流程，資料集的部分就是看從db或是網路上爬蟲爬下來都可以，再來步驟2資料清洗是將資料變成電腦能計算的格式，步驟3特徵工程就是前面說人工給特徵的部分，這兩個我們統稱為資料前處理，也是今天的主題，再來等數據都處理好特徵都找好後，我們就可以將資料丟給模型訓練，而為了能讓模型在沒新的沒見過的資料上也表現良好，我們會將資料切成訓練用的與測試用的，訓練用的去餵給模型訓練，然後再用測試用的資料去評估模型的效能，測試效能ok後，我們才會將實際的資料丟到模型去做預測

8.

接下來就是今天的正題資料前處理，不管事做甚麼機器學習的任務，我們的第一步驟都是前處理，下面有兩個統計圖。左邊的是資料科學家或資料分析師在專案中不同步驟所花的時間，右邊則是各個步驟中哪個最討人厭，有趣的是兩張圖的前兩名都一樣，第一名都是前處裡，第二名則都是資料收集，可以發現專案大部分時間都在處理這兩件事，反而]建模與分析只占少部分時間

9.

而這邊這兩句話算是在機器學習領域很常聽到的名言，他們也很好的說明了為什麼資料前處理這麼重要，兩者都在描述同一件事，就是資料與特徵的重要性，舉個簡單的例子，假如今天我們要做一個辨認男女的二元分類問題，你給模型1000張照片，選擇將膚色瞳孔顏色作為特徵，不管甚麼算法都無法良好的區分男女，但如果我們是選染色體為特徵，娜隨便選個算法都會比前一種好

10.

接下來簡單介紹一下我們前處理常用的工具，因為我本身是寫python比較多，這邊就用python來跟大家介紹，首先是Numpy，他是python在進行科學運算得時候最基礎的package也是最核心的，其他有許多package都是在numpy的基礎上發展出來的，它的優點有下面幾點，簡單來說就是能方便計算多維度的陣列，函有許多數學方程式可以用，以及資料型別轉換方便之類的，相關的操作語法網路上很好找，也不會很難，有興趣的可以去玩看看

11.

再來就是PANDAS，有些人會叫他PYTHON的EXCEL試算表，他沒有EXCEL對資料列數與維度的限制，並結合了NUMPY與SQL的操作能力，讓操作大量資料變得比較簡單一些，然後他比較特別的資料型態有series 和dataframe兩個，series有點像帶有index的array，而dataframe有點像函有一堆series的字典，然後結構化操作像是重組切割聚和子集和之類的都能用簡單的語法完成

12.

最後是scikit-learn簡稱sklearn，他是開源的python機器學習包，裡面基本上涵蓋了所有主流的演算法，在實際專案的執行中，手寫出一個演算法的機會比較少，除了耗時外，架構是否清晰與穩定性的強弱都是問題，因此比較多時後是根據資料的特徵選擇演算法，利用像sklearn這類的package去呼叫算法，並針對結果去調整算法的參數

剛剛講的三個package操作不難網路上都有很多資料，有興趣的可以去丸看看

13.

講完使用的工具，接下來是我們資料前處理的流程，這邊是我自己去過歸納的，可能步驟的數量或名稱與網路上的資料有些不異樣，但內容都差不多相同，清洗與轉換主要是讓演算法能夠計算，特徵篩選與抽取則以不同的方式留下有用的特徵，後面會做說明

14.

接下來是一些資料常見的問題，假如這邊是我們要清洗的資料，紅色的部分代表缺失的資料，藍色代表格式不統一的資料，黃色代表不準確的資料，像是這欄應該只有N或Y兩種格式，而綠色代表重複的資料，剛剛說的這些都是比較好解決的，replace一下或是直接刪除都可以，接下來講一些比較慣用的手法

15.

首先對缺失值的處理，最簡單的方法是刪除，這概念就是刪掉缺失的部分我們就沒有缺失的問題，當然 這個方法有比較多的限制，假如資料集中含有缺失值的部分只佔整個資料集的一小部分，直接刪掉當然沒問題，但如果現在全部資料的99%都有或多或少的缺失值，全刪掉就不可行

下面一個小範例，可以發現資料裡面有空值Nan，我們可以利用pandas內建的函數對她做刪除，像是有空值就刪除，按照行或列將資料刪除，所有特徵階為空值才刪除，依照空值出現次數刪除，或是指定刪除類別之類的

16.

這邊額外補充一下缺失數據的種類，簡單的說明在上面，這邊舉一些例子比較好理解，完全隨機缺失就像假如我們做問卷調查，但某一題我們用擲骰子來決定是否回答，是否回答與其他問題沒有相關

隨機缺失像我們問收集了1000個人的身高體重年齡，發現體重缺失值30%，一查下去發現只要是30歲以下的女生體重就不回答，這情況的缺失值就跟其他變數有關

最後非隨機缺失，一樣問卷調查年收工作之類的，發現有20%的人年收是缺失值，並且我們從其他變數看也找不出原因，因此年收的缺失與否可能與他值的高低有關，也就是只與自身有關

前兩種可根據情況刪除缺失資料，隨機缺失甚至可以通過其他便量去進行估算

而非隨機缺失若直接刪除缺失值則可能導致模型出現偏差，對缺失值進行填補也比較麻煩

17.

接下來介紹缺失值處理的第二種辦法，找一個數字把缺失的地方補上，而要用甚麼數字補，可以透過一些模型攻勢或是自己決定，首先是最基本的用中位數和平均數進行填補，他在python裡有寫好的function可以直接用

這邊教大家比教簡單的方式去判斷怎麼去決定用哪個補植，若資料的分布是常態分佈的話用平均數或中位數都可以，但如果資料是偏態分布或是傾斜的話，力用中位數去填補比較適合，接下來跟大家介紹一夏資料的分布

18. https://zhuanlan.zhihu.com/p/72398933

這張圖大家以前上學的時候應該都有看過，這種圖叫正態分布也較高斯分布，在機器學習里，越簡單的模型越常用，原因就是她很好被理解與解釋

這邊幫大家簡單複習一下相關的概念，假如我們現在要做一個預測模型，要去精準的預測某個變數的值，首先要做的事情就是了解這個變數的特性，舉例來說，假如我要去預測下次丟骰子可能的值，要先知道她可能值的範圍是在1到6，再來就是去確定這1到6的選項中每個值發生的概率，像是骰子只有六個面，他出現7的機率就是0，

實際操作中就是我們去重複的丟這個骰子，去紀錄每個可能結果發生的次數，有了這些資料後我們就可以去畫出概率分布的取縣就像上面的圖異樣，知道變數的分布後，接著就可以開始去估計每個結果出現的機率

而正態分布是所有概率分布中最常使用的類型，像是人的身高 每天回家所花的時間或是線性回歸中的殘差值基本上都比較接近正態分布

而正態分布的圖繪呈現一個鐘形的曲線，而他只依賴資料集中的兩個特徵，平均值和方差，平均值應該大家都知道，方差指的是衡量樣本總體偏離平均值的程度，正態分布這個統計的特性會讓預測問題變得很簡單，任何具有正態分布的變數，基本上都可以進行高精準度的預測

正態分布很容易去解釋的原因有兩個，他的平均值眾數和中位數都異樣，再來就是用平均值和標準差就可以去理解整個分布

而為甚麼正態分布這麼常見，這就會扯到一堆數學像是中心極限定理和一些概率分布函數，這個有興趣的可以上網去找資料看一下

18.

了解為什麼許多機器學習的模型都很喜歡用正態分布後，我們就可以去理解為什麼要處理非正態分布也就是偏態分不得資料了，偏態分部的突有兩種，分為副偏態和正偏態，而偏移的程度會用偏度這個統計量去衡量，圖上的名詞MODE指的是眾數MEAN是平均MEDIAN是中位數，塗上可以發現，中位數所在的位置是最接近正態分布的中心點，因此回到前面講的，當我們遇到資料傾斜成偏態分布時，用中位數去做缺失值的填充，會讓我們資料的分布更往正態分布的方向移動

18.

另外，除了偏度之外，還有一個統計輛鋒度也可以衡量資料是否為正態分布，這也分為尖峰和低鋒兩種，圖尚可以看出像尖峰分布這種類型，他的平均值也就是他鋒的地方和他尾巴的地方也就是正副兩端發生的機率都比正態分布高，而低鋒分布就正好相反，通常我們會拿偏度和鋒度搭配去衡量資料分布的狀態

19.

理論的東西說完後，接下來講實際操作的部分，