Pre-processing:

在建立模型以前，對資料做預先處理是非常重要的。原始數據中的異常值、數值範圍、資料缺失等因素可能會很大程度地影響分類器的模型表現。讓我們來一個一個討論是否需要執行以上三種預先處理技術(以下統計數據皆經由Excel計算得出)：

1. outlier detection：在常態分佈下，異常值檢測的一個常用方法是檢測該數據集中是否有數據位於平均數三個標準差開外。首先，我們先計算所有屬性的標準差(在本proposal中，均以前10個屬性為例，若需所有原始數據的統計資料，可見附件)：



第二，我們可以先行簡單地用MIN、MAX、AVERAGE函數來獲取每個屬性的最大值、最小值以及平均數：





將以上四行數據進行比較，毫無疑問地，每一筆資料都存在著異常值，這明顯可能會影響到分類器的表現。不過直接用肉眼觀察可能不夠嚴謹，讓我們再次使用Excel中的函式：

=SUMPRODUCT((A2:A1601 > AVERAGE(A2:A1601) + 3 \* STDEV.P(A2:A1601)) + (A2:A1601 < AVERAGE(A2:A1601) - 3 \* STDEV.P(A2:A1601)))

這個函式將由Excel計算。將計算每個屬性中總共有幾個異常值存在。以下是前十筆資料的計算結果：



事實上不僅僅是前十個屬性，可以說所有屬性都有異常值的發生。因此，儘管我們仍需透過交叉驗證來決定是否為原始資料進行異常值檢測。但我認為，對原始資料進行異常值檢測是相當有必要的。

至於對異常值的處置方式，一般來說通常有兩種：刪除該筆資料，或是將異常值改為平均數。在決定處理方法之前，我們可以先計算出每個屬性的異常值發生機率：



以及其最大與最小發生機率：7.25% / 4.375%。

一般來說，異常值為7.25%~4.375%的比例屬於可接受範圍內，是可以刪除異常值發生的該筆資料的。儘管如此，為了更好的分類器，還是可以對其進行交叉驗證來選出對異常值最佳的處置方式。

1. Normalisation：資料的歸一化用於避免某個屬性其權重遠大於其他屬性的情況。通常發生於某屬性的domain特別大的時候。我們同樣可以先利用Excel來獲取每個屬性的數值domain(由屬性最大值 – 屬性最小值產生)：



計算完成後擷取出此1600個屬性的最大與最小domain：1371.19、0.095。

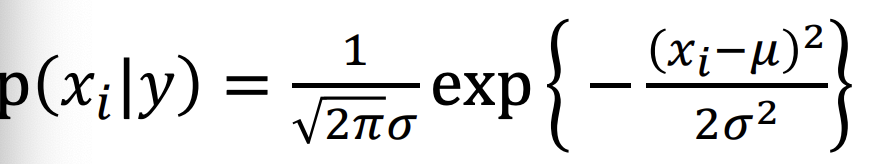
經由以上，我們可以得出資料的歸一化也是相當重要的，否則屬性值domain越大的(例如1371.19)可能會主宰分類器邏輯，而越小的(例如0.095)可能會被分類器邏輯忽視，但這並不代表他們不重要或是特別重要。每個屬性應當要有公平的權重。

1. Imputation：Imputation的處置較為simple。對於所有有缺失的資料，我們有幾種處置方式：直接忽視、插入所有資料的平均值、插入與該筆資料同一個class的該屬性平均值。直接忽視的方法顯然我們也可以”直接忽視”，但對於剩下兩種方法，乍看之下也許插入同一class的該屬性平均值為最優解。但事實上我們仍需對此兩種方法進行交叉驗證用以確定哪一種處置方法可以使得分類器運作地更好。

Application of Classification Techniques：

從第二週至第八週的教學內容我們可以得出decision tree、random forest、k-nearest neighbour and naïve bayes這四種方法是此case之下的潛在分類器邏輯。沒有任何一種方法是壞方法，重要的是我們要透過多次交叉驗證來確定使用什麼方法以及其hyperparameter。此外若條件允許，對於任何分類方法我們都應該要保持80%樣本作為訓練樣本，20%作為測試樣本。

* Decision tree：利用屬性將資料做分組，無論是root node或是non-root node都有屬於自己的停止標準，如果沒有達到停止分類標準就將繼續分類下去。Decision tree的運作邏輯相當好理解。其難點為要用哪個屬性來作為下一個分類標準，在這之中就會牽涉至交叉驗證。首先選定任意課堂上提及的purity measure(Information gain, Gain ratio, Gini index)作為判定purity的方法。接著，因為此資料集中的所有資料都是連續數字。因此，使用同樣課堂上提到的Continuous values將每個屬性分為兩組數字，從每個屬性中挑選出最高(或最低，depends on你剛才選擇的purity measure)的指數與其他屬性最高(或最低)的指數做比較，從高到低(或從低到高)排序即為分類標準的順序。
* Random forest：Random forest雖然與Decision tree都是以”tree”的樣子來進行分類，但是兩者的運作邏輯大有不同。在這個case中，Random forest會從1600\*80% = 1280筆資料中隨機選取1280筆資料(允許重複)作為訓練樣本。依照慣例，我們預設將創建100棵樹，每棵樹將分配到128個資料子集(同樣可重複)。對其進行正常decision tree的建構。建構完的森林將對unseen的新資料進行投票表決，意即這筆資料會被放入100棵樹中進行分類，根據每顆樹的分類結果進行投票已決定此新資料的最終分類。在建構Random forest期間也會涉及到交叉驗證，即創建單一決策樹的時候該採用何種purity measure(如同第一種方法一樣)。
* 𝑘-Nearest Neighbours：在此方法中，有兩個特別重要的因素需要考慮。也就是該使用何種距離衡量方法來評斷點與點之間的距離。根據此case的描述，若有某一個點她的第106個屬性為1，也就是這個基因具有細胞通訊功能。那麼越靠近此點的其他資料點理應來說在分類上應與此點越接近。即若兩點距離相當近僅有些許數值差別，應該將其分類為同一個類別而非不同類別。根據以上敘述，我們在此專案中可能採取曼哈頓距離。至於k值的決定，除了遵守老師課堂中提到的”k數值一般情況下需要為奇數”外。我們還需要透過交叉驗證來決定哪一個奇數。首先，將1280筆training data分為多個擁有相同數量資料點的資料夾(取決於計算成本)。選定其中一個資料夾為testing data，其餘為training data。對於testing data中的每一個點，計算該點與所有training data的距離(曼哈頓距離)。接著，對於每一個testing data的點，令k = 1, 3, 5……(取決於計算成本)獲取與該testing data距離最近的k個點，對於這些k個點，根據ground truth中的training data的class進行多數決投票得出k = n時該資料點預測應為哪一個分類而得出prediction table。最後，對於testing data的prediction table，比較prediction table的預測結果與ground truth得出k = n時的準確率(或是F1 score)。選出預測表現最好的k值。
* Naïve Bayes：對於全是連續小數的這個case來說，使用Naïve Bayes會需要大量的計算。但是邏輯也相對單純。我們只需要透過老師在上課提到的公式：



任何新資料進入後，令y為1(第一個class)。遍歷所有屬性x。xi為當前屬性的，將後面的公式更改為xi屬性的對應統計數值，計算出p(xi | y), i = 1, 2, 3, 4, …, j各為多少之後相乘，得到的數字與令y = 0(第二個class)且後續計算方法相同得出的數字後做比較。較高的即為模型預測答案。顧名思義，Naïve Bayes是屬於比較簡單單純的計算方式，因此在設計模型內部時並不特別考慮hyperparameter。