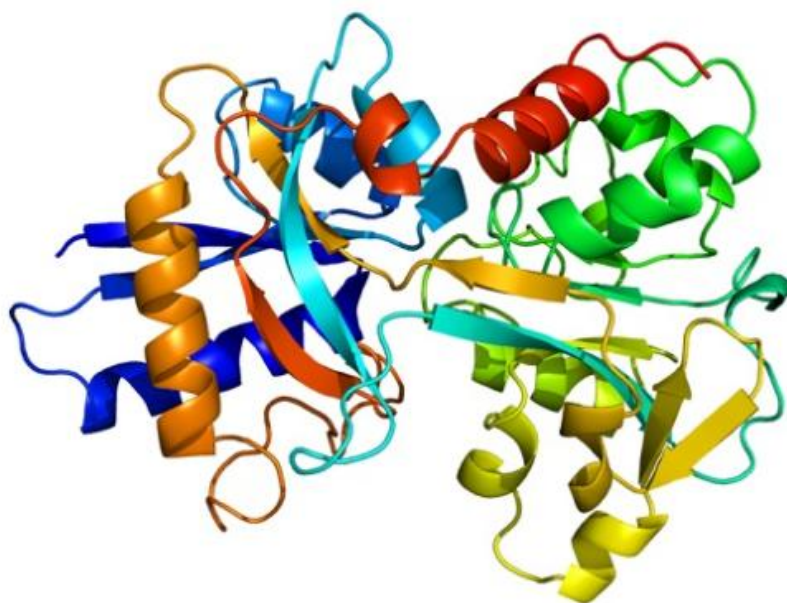


國立陽明交通大學資訊管理研究所
資料探勘研究與實務

結構蛋白質氨基酸序列分類預測



第四組 成員

11034005 許秀琪

11034006 許秀琳

10934022 徐嘉

409706007 林德全

授課教師:劉敦仁 教授

摘要

本專題研究採用Kaggle平台公開資料集(結構蛋白質氨基酸序列資料集, protein DNA sequence dataset), 來進行資料集分析與建模。我們在資料集分析中瞭解此資料集總共有4989種蛋白質類別, 各分類樣本數量不平均, 故分析中我們只取前五大分子類別來建立預測模型。

運用本次課程所學習的機器學習及深度學習模型, 來進行模型訓練及評估。其中實驗步驟包括資料前處理, 特徵工程, 多種模型搭建與訓練。最後我們在實作結果進行各模型訓練結果之評估, 以瞭解目前使用機器學習與深度學習的模型預測成效是否有顯著的差異, 以及其優缺點為何。

透過本專題研究實作, 我們可以將課程所學習之分析與建模方法, 進行充份的練習, 對機器學習理論與工具能更加的瞭解。

關鍵字: 資料探勘、機器學習、深度學習、蛋白質結構。

一、前言

蛋白質結構有助於設計抗體或藥物，以前要花很多人力及時間來建構出蛋白質結構，現在若有準確的預測模型，有助於減少研發成本。未來如果能研究出如何折疊的原理就可能直接人工創造出新的酵素。

蛋白質的分子結構可分為四級，簡要描述如下：

蛋白質一級結構：組成蛋白質多肽鏈的線性胺基酸序列。(*為本專題所研究的結構)

蛋白質二級結構：由一級結構序列依靠不同胺基酸之間的C=O和N-H基團間的氫鍵形成穩定的二級結構。

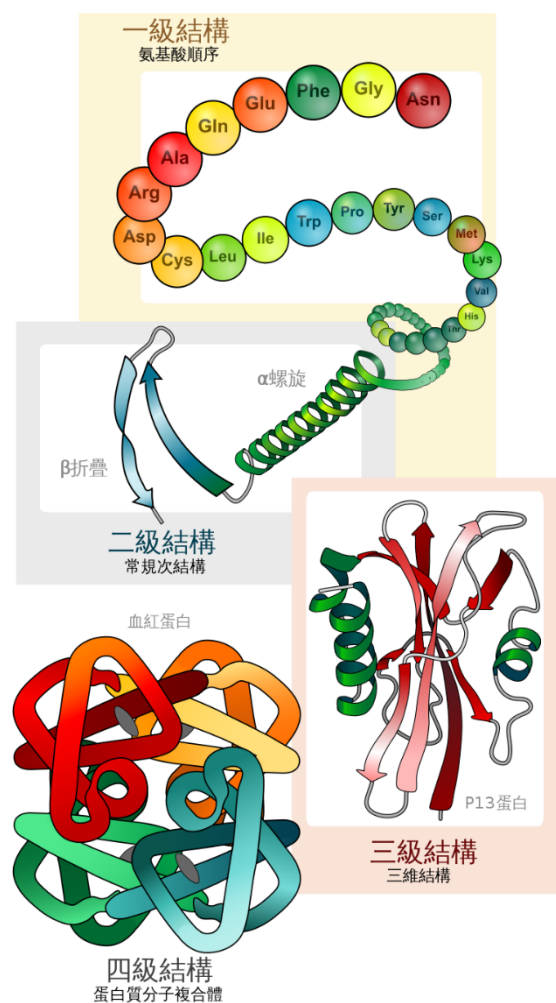
蛋白質三級結構：通過多個二級結構元素在三維空間的排列所形成的一個蛋白質分子的三維結構。

蛋白質四級結構：用於描述由不同多肽鏈(亞基)間相互作用形成具有功能的蛋白質複合物分子。

([4] 參考wiki data)

本專題預計解決問題：

利用結構蛋白質胺基酸序列資料集來建立**分類預測模型 (蛋白質一級結構)**，這是2018年在kaggle平台被提出的問題，我們希望能利用這兩年新發表的NLP深度學習模型(例如:RNN, LSTM, Transformer, BERT.. e.g.)來提升這個問題的分類預測準確度。



二、實作方法

2.1 機器學習模型介紹

- **Naive Bayes**

樸素貝葉斯演算法是應用最為廣泛的分類演算法之一。它是基於貝葉斯定義和特徵條件獨立假設的分類器方法。NB模型所需估計的引數很少，對缺失資料不太敏感，演算法也比較簡單。由於樸素貝葉斯法基於貝葉斯公式計算得到，有著堅實的數學基礎，以及穩定的分類效率。

貝葉斯公式是英國數學家提出的一個資料公式：

$$p(A|B) = \frac{p(A, B)}{p(B)} = \frac{p(B|A) \cdot p(A)}{\sum_{a \in \mathcal{F}_A} p(B|a) \cdot p(a)}$$

$p(A, B)$: 表示事件A和事件B同時發生的概率。

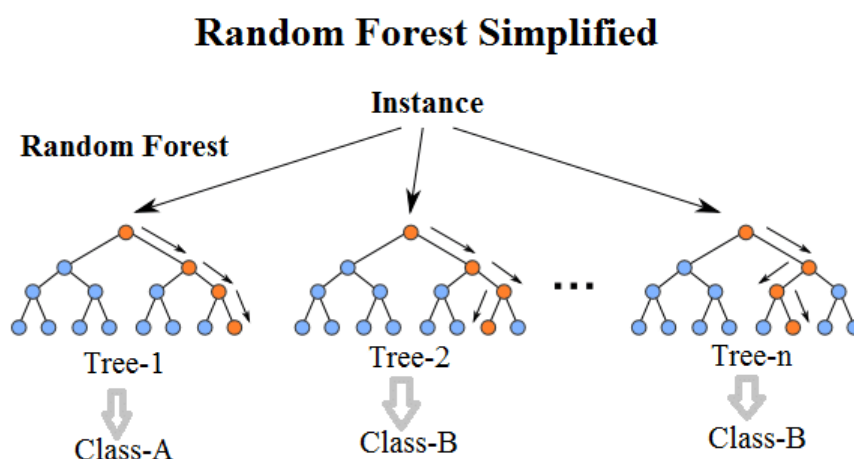
$p(B)$: 表示事件B發生的概率，叫做先驗概率； $p(A)$: 表示事件A發生的概率。

$p(A|B)$: 表示當事件B發生的條件下，事件A發生的概率叫做後驗概率。

$p(B|A)$: 表示當事件A發生的條件下，事件B發生的概率。

我們用一句話理解貝葉斯：世間很多事都存在某種聯絡，假設事件A和事件B，人們常常使用已經發生的A事件去推斷B事件的概率。Naive Bayes假設所有特徵的出現相互獨立互不影響，每一特徵同等重要，又因為其簡單，而且具有很好的可解釋性一般。相對於其他精心設計的更複雜的分類演算法，樸素貝葉斯分類演算法是學習效率和分類效果較好的分類器之一。樸素貝葉斯演算法一般應用在文字分類，垃圾郵件的分類，信用評估，釣魚網站檢測等。

- **RandomForest**



隨機森林是一個包含多棵決策樹的模型，在森林裡面建構一棵棵各自獨立的決策樹，最後以投票方式(眾數/取平均)來決定最終的結果。

最主要的運作原理為Bagging，採取取後放回的方式建立資料子集，並用這些不同的資料子集來建立森林裡的決策數。隨機森林採用Bootstrap的方式分別對樣本以及特徵進行取後放回的抽樣，建立起一棵棵的決策樹。在兩個隨機因子之下，讓隨機森林較不容易產生過度配適的現象。當森林裡的決策樹都建構好後，最終將以投票的方式來決定結果。對於離散型的資料，將採取個別決策樹結果的眾數；對於連續型的資料，則採取個別決策樹結果的平均值。

隨機森林可以處理的資料集非常廣泛，可處理連續型資料亦可處理離散型的資料，更可以處理高維度的特徵資料。此外，在兩種隨機因子的抽取下，更可以讓隨機森林不容易產生過度配適的結果。另一方面，過多的決策樹容易導致計算成本的提高，包含時間與空間的成本。另外，若資料本身的雜訊過多，還是會讓隨機森林出現過度配適的結果。

● Adaboost

Adaboost是一種迭代算法，其核心思想是針對同一個訓練集訓練不同的分類器(弱分類器)，然後把這些弱分類器集合起來，構成一個更強的最終分類器(強分類器)。Adaboost算法本身是通過改變數據分佈來實現的，它根據每次訓練集之中每個樣本的分類是否正確，以及上次的總體分類的準確率，來確定每個樣本的權值。將修改過權值的新數據集送給下層分類器進行訓練，最後將每次得到的分類器最後融合起來，作為最後的決策分類器。

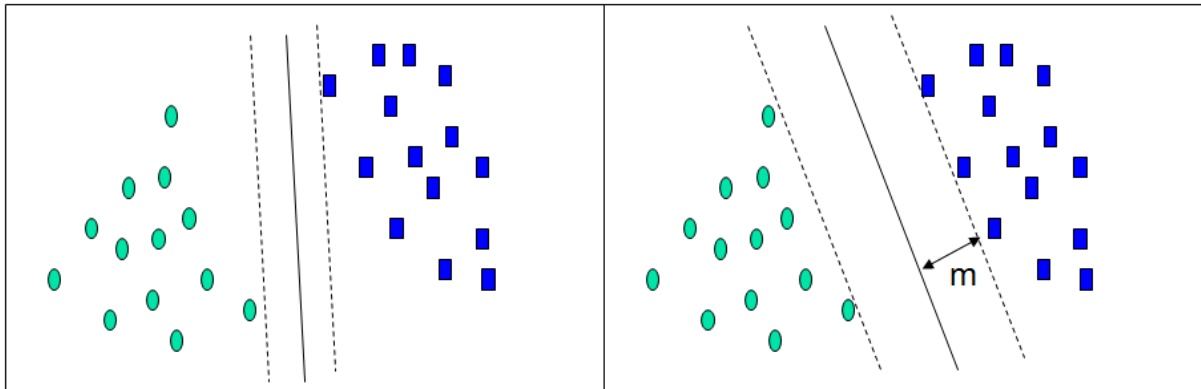
算法步驟：

1. 給定訓練樣本集S，其中X和Y分別對應於正例樣本和負例樣本；T為訓練的最大循環次數；
 2. 初始化樣本權重為 $1/n$ ，即為訓練樣本的初始概率分佈；
 3. 第一次迭代：(1)訓練樣本的概率分佈相當，訓練弱分類器
(2)計算弱分類器的錯誤率
(3)選取合適閾值，使得誤差最小
(4)更新樣本權重
- 經T次循環後，得到T個弱分類器，按更新的權重疊加，最終得到的強分類器。

● SVM

支援向量機(Support Vector Machine, SVM)是一種基於統計學習理論基礎的機器學習模型，針對小樣本、非線性、高維度與局部最小點等問題具有相對的優勢。這個概念其實早在1960-1990年代就由數學家Vapnic及Chervonenkis等人所提出，並建立了這套統計學習理論。除了在文字分類、圖像分類及醫學中分類蛋白質等領域有不錯的成效外，因具有計算速度快且空間成本低等優勢，在工業界也有廣泛的應用。

SVM是一種線性分類器，同時卻也可以推展到解決非線性的分割問題。具象化來說，SVM就是將在低維度空間線性不可分的樣本映射到高維度空間去，找到一個超平面將這些樣本做有效的切割。



2.2 深度學習模型介紹

- **1D-CNN**

卷積神經網絡(Convolutional Neural Network)簡稱CNN，當我們說CNN時，通常是指用於圖像分類的2D-CNN。而一維卷積神經網絡只在一定程度上有所涉及，比如在自然語言處理(NLP)中的應用。CNN 可以很好地識別出數據中的簡單模式，然後使用這些簡單模式在更高級的層中生成更複雜的模式。當你希望從整體數據集中較短的(固定長度，即kernel size)片段中獲得感興趣特徵，並且該特性在該數據片段中的位置不具有高度相關性時，1D CNN 是非常有效的。1D CNN 也可以很好地應用於傳感器數據的時間序列分析(比如陀螺儀或加速度計數據)；同樣也可以很好地用於分析具有固定長度週期的信號數據(比如音頻信號)。

- **LSTM**

LSTM(Long short-term memory)，是目前RNN(Recurrent Neural Network)中最常使用的模型。原始的RNN在訓練中，隨著訓練時間的加長以及網路層數的增多，很容易出現梯度爆炸或者梯度消失的問題，導致無法處理較長序列資料，從而無法獲取長距離資料的資訊。它主要由四個Component組成：Input Gate、Output Gate、Memory Cell以及Forget Gate。LSTM應用的領域包括：文字生成、機器翻譯、語音識別、生成影像描述和視訊標記等。

- **Transformer**

Transformer模型是一種神經網路，藉由追蹤序列資料中的關係，學習上下文之間的脈絡及意義，就如同一個句子中的每一個字。它是使用一套不斷發展，稱為注意力(attention)或自我注意力(self-attention)的數學技術，它可偵測一個系列中以微妙方式相互影響和相互依賴的資料元素，甚至是模糊的資料元素。Transformer模型由Google在2017年在Attention Is All You Need[1] 中提出，

該文使用Attention 替換了原先Seq2Seq模型中的循環結構，給自然語言處理(NLP)領域帶來極大震動。Transformer模型是迄今發明出最新且最強大的模型之一，有些人稱其為 Transformer人工智慧。隨著研究的推進，Transformer 等相關技術也逐漸由NLP 流向其他領域，例如計算機視覺(CV)、語音、生物、化學等。

2.3 相關分析工具

通常機器學習模型搭建，都會使用模型解釋工具來瞭解模型從資料中學習到什麼特徵，而特徵的重要性與形態是如何。我們在本專題中使用SHAP (SHapley Additive exPlanations) [7] 來分析模型學習到的特徵狀況。

本專題資料集共有五個分類類別，Class 0,1,2,3,4 其代表的名稱如下：

Class List: 各類別編號代表的名稱

Class 0: 核糖蛋白(Ribosome)

Class 1: 水解酶(Hydrolase)

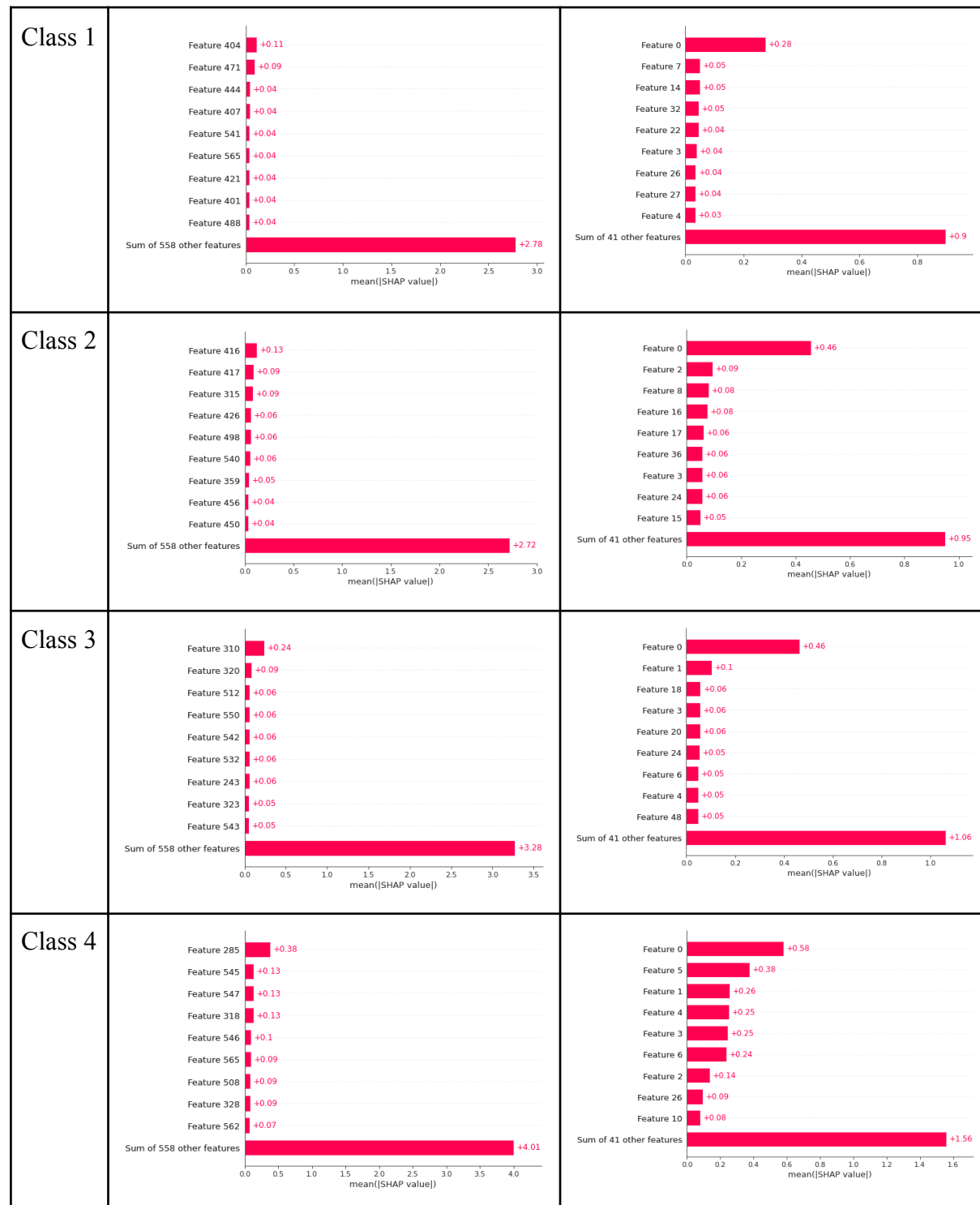
Class 2: 轉移酶(Transferase)

Class 3: 氧化還原酶(Oxidoreductase)

Class 4: 免疫系統(Immune system)

我們利用SHAP Value來比較兩種特徵工程的特徵重要度分佈狀況，在下表中分別以各類別編號來呈現這個類別在分類時，有那些特徵是比較重要的(top 10)特徵。其中可以看出在以sequence位置編號下有567個特徵，它的特徵重要性都很均勻，在0.1~0.5的SHAP value，並沒有特別明顯的重要特徵，也就是所有特徵都對模型做出正確分類有一點供獻。反觀在使用PCA-50特徵組合下，第一組(Feature 0)特徵對模型分類有明顯重要性(相對於其它特徵)，這符合PCA在演算上第一組特徵向量(eigenvector)具有較高的模型解釋能力。

	Feature size: 567 (tokenizer)	Feature size: 50 (PCA)
Class 0	<p>Feature 356 +0.53</p> <p>Feature 566 +0.19</p> <p>Feature 563 +0.17</p> <p>Feature 550 +0.14</p> <p>Feature 564 +0.13</p> <p>Feature 270 +0.12</p> <p>Feature 560 +0.1</p> <p>Feature 548 +0.09</p> <p>Feature 408 +0.09</p> <p>Sum of 558 other features +4.81</p> <p>mean(SHAP value)</p>	<p>Feature 0 +1.19</p> <p>Feature 7 +0.22</p> <p>Feature 2 +0.21</p> <p>Feature 3 +0.16</p> <p>Feature 1 +0.14</p> <p>Feature 5 +0.13</p> <p>Feature 4 +0.11</p> <p>Feature 10 +0.11</p> <p>Feature 6 +0.08</p> <p>Sum of 41 other features +1.77</p> <p>mean(SHAP value)</p>



三、實作步驟

3.1 資料探勘流程說明

資料探勘是一種分析大量資料的方法，其目的是找出這些資料中隱藏的規律和模式。這些規律和模式可能有助於預測未來的趨勢，並幫助決策者做出更明智的決策。

資料探勘通常使用軟體工具和統計學方法來幫助分析和理解資料。而資料探勘的實作流程有一個系統性的流程方法可以參考，如圖3.1 資料探勘流程，我們先透過問題討論來瞭解問題有關的資料來源，並進行資料收集(或公開資料集找尋)，這些收集到的資料集就是我們的輸入資料(Input_Data)，而資料本身因為各種取樣因素，會存在雜訊或缺失(遺失)資料，所以在資料輸入後，我們要進行資料前處理作業(Data_Preprocessing)，其中包括特徵選擇(Feature_Selection)選出對預測結果有重要影響的特徵，利用維度縮減(Dimensionality_Reduction)來減少特徵數量，讓模型架構複雜度減低，以降低雜訊干擾與過度配適(Overfitting)的問題。另外利用資料標準化(Normalization)來控制特徵值域範圍，一方面可以降低雜訊干擾，亦可加速模型收斂速度。再來是利用資料集切分(Data_Subsetting)將原始資料集切分為訓練集(Training set)及測試集(Testing set)，訓練過程會在訓練集中再分出驗證集來做訓練過程的模型成效驗證。

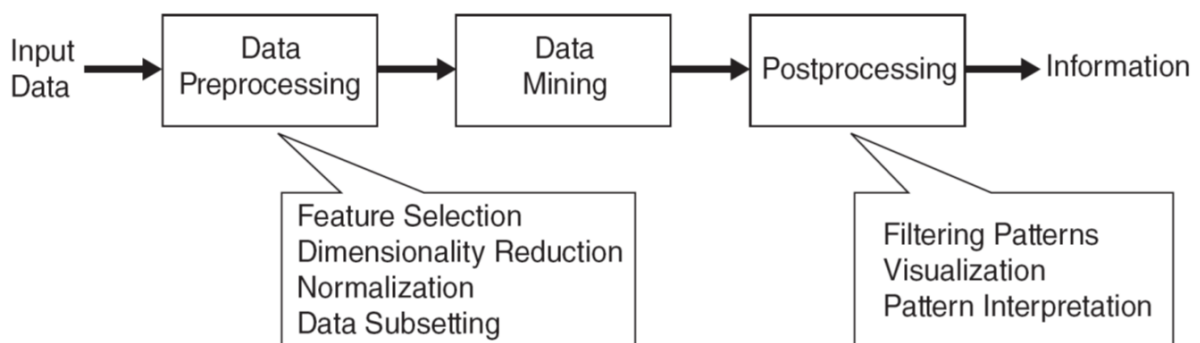


圖3.1 資料探勘流程

在流程中的資料探勘(Data Mining)步驟，我們會使用各種機器學習模型來進行初步模型的訓練。後處理(Post processing)步驟中我們會進行模式過濾(Filtering Patterns)，訓練成果視覺化(Visualization)，模式解釋(Pattern Interpretation)及獲取推論資訊(Information)。

3.2 資料集敘述

資料來源網址(<https://www.kaggle.com/shahir/protein-data-set>) Structural Protein Sequences - Sequence and meta data for various protein structures, 此資料集是從Research Collaboratory for Structural Bioinformatics(RCSB) 及Protein Data Bank (PDB)兩家機構取得的部份資料, 其中包括項目欄位如下表所示:

表3.1 資料集欄位說明表

欄位名稱	型別	欄位說明
structureId	object	分子結構ID (Primary Key)
classification	object	分子類別 (本專題的Label Y)
experimentalTechnique	object	實驗的使用技術
macromoleculeType	object	大分子類型
resolution	float64	3D結構的影像解析度
structureMolecularWeight	float64	結構分子量
crystallizationMethod	object	結晶方法
crystallizationTempK	float64	結晶的溫度
densityMatthews	float64	晶格密度
densityPercentSol	float64	結晶溶劑比率
pdbxDetails	object	蛋白質資料庫的詳細內容
phValue	float64	結晶pH值
publicationYear	float64	發佈年份
chainId	object	分子鏈ID
residueCount	object	蛋白質殘基數量
sequence	object	蛋白質氨基酸序列

1. 檔案說明:

資料集共有兩個檔案:

- **Pdb_data_no_dups.csv**
主要記錄各組蛋白質發佈時間, 實驗方法及蛋白質分類, 等 Meta data。
- **Pdb_data_seq.csv**
依 structureId 分別記錄各蛋白質氨基酸序列(sequence)資料。

2. 蛋白質氨基酸序列(sequence)說明:

蛋白質氨基酸序列通常表示為字母串, 三個字母代碼或單個字母代碼可以用於表示 20種天然存在的胺基酸, 以及混合物或不確定的胺基酸 (類似於核酸符號)。(參考 wiki)

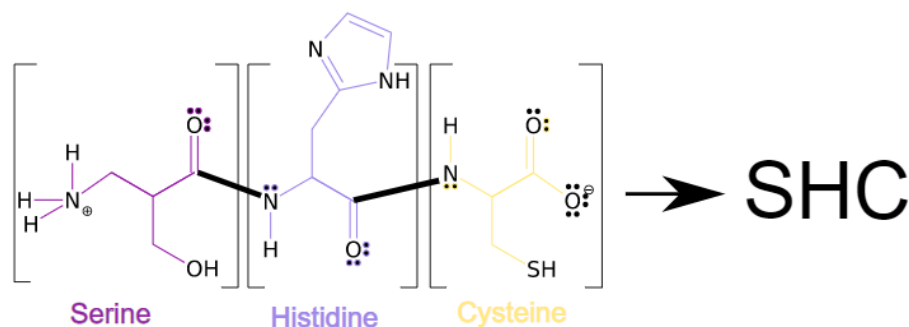


圖3.2 蛋白質氨基酸序列表示示意圖 [6]

- 分析方法

1. 資料預處理:

匯入 pdb_data_no_dups.csv 檔案, 計有 141,401 筆資料。(Meta data檔)

匯入 pdb_data_seq.csv 檔案, 計有 467,304 筆資料。(sequence檔)

其中 Meta data 有兩筆 classification 為 NaN(not a number), 故需先去除。

	structureId	classification	experimentalTechnique	macromoleculeType	residueCount	resolution
113250	4WM2	NaN	X-RAY DIFFRACTION	Protein	129	1.6
138421	5VZL	NaN	ELECTRON MICROSCOPY	Protein#RNA	1574	3.9








之後, 我們利用 structureId 這個共通欄位來合併 Meta data 及 sequence 檔案。

2. EDA (Exploratory data analysis):

缺失值狀況分析: 共計有9個欄位有缺失值的狀況, 所幸本專題做為輸入變數(Input X)的 sequence 欄位沒有缺失值, 故缺失值可以先不予處理。

Columns	% of Missing Data
0 crystallizationMethod	33.2
1 crystallizationTempK	32.6
2 pHValue	27.7
3 pdbxDetails	18.2
4 densityMatthews	17.3
5 densityPercentSol	17.3
6 publicationYear	12.2
7 macromoleculeType	7.6
8 resolution	4.6

分析 classification 欄位之數值次數分佈：

Value	Count	Frequency (%)	
RIBOSOME	60710	12.9%	
HYDROLASE	47833	10.2%	
TRANSFERASE	37726	8.0%	
OXIDOREDUCTASE	35114	7.5%	
IMMUNE SYSTEM	15989	3.4%	
LYASE	11871	2.5%	
HYDROLASE/HYDROLASE INHIBITOR	11262	2.4%	

可以得知總共有 4989 種不同類型的蛋白質類別，其中前五大類別為：

1. RIBOSOME(12.9%) - 核糖蛋白(ribosome)
2. HYDROLASE(10.2%) - 水解酶(Hydrolase)
3. TRANSFERASE(8.0%) - 轉移酶(Transferase)
4. OXIDOREDUCTASE(7.5%) - 氧化還原酶(Oxidoreductase)
5. IMMUNE SYSTEM(3.4%) - 免疫系統(Immune system)

名詞解釋:(參考 wiki)

- 1) 核糖蛋白(**ribosome**): 舊稱「核糖核蛋白體」或「核蛋白體」，是細胞中的一種胞器，由一大一小兩個次單元結合形成，主要成分是相互纏繞的RNA (稱為「核糖體RNA」，ribosomal RNA, 簡稱「rRNA」) 和蛋白質 (稱為「核糖體蛋白質」，ribosomal protein, 簡稱「RP」)。
- 2) 水解酶(**Hydrolase**): 是一種催化化學鍵的水解的酶。
- 3) 轉移酶(**Transferase**): 是一種催化一個分子 (稱為供體) 的官能團 (如甲基或磷酸鹽團) 轉移至另一個分子 (稱為受體) 的酶。
- 4) 氧化還原酶(**Oxidoreductase**): 是一種催化電子由一個分子傳送往另一個分子的酶。
- 5) 免疫系統(**Immune system**): 是生物體體內一系列的生物學結構和進程所組成的疾病防禦系統。

我們以這五大類別來做為模型的分類標籤(Label Y)

	class	F	T	R
0	RIBOSOME	410435	60710	0.1289
1	HYDROLASE	423312	47833	0.1015
2	TRANSFERASE	433419	37726	0.0801
3	OXIDOREDUCTASE	436031	35114	0.0745
4	IMMUNE SYSTEM	455156	15989	0.0339

合計197,372筆資料, 位原本資料集的41.9%。(T:筆數, R:佔全體比率)

而其它分類類別筆數資料偏少, 不利建立準確的模型, 故先不計入。

3.3 特徵工程

a. 特徵抽取: 使用n-gram range(min:4, max:4)

analyzer: char_wb (依sequence text 取最大/最小長度的字元)

```
vect = CountVectorizer(analyzer = 'char_wb', ngram_range = (4,4))
```

產生的範例:

```
['yyya', 'yyyc', 'yyyd', 'yyye', 'yyyf', 'yyyg', 'yyyh', 'yyyi', 'yyyk', 'yyyl', 'yyym', 'yyyn', 'yyyp', 'yyyq', 'yyyr', 'yyys', 'yyyt', 'yyyv', 'yyvw', 'yyyy']
```

評估n-gram適當的裁切長度

- range(2,2)長度產生範例:

```
['yc', 'yd', 'ye', 'yf', 'yg', 'yh', 'yi', 'yk', 'yl', 'ym', 'yn', 'yp', 'yq', 'yr', 'ys', 'yt', 'yv', 'yw', 'yx', 'yy']
```

- range(3,3)長度產生範例:

```
['yyc', 'yyd', 'yye', 'yyf', 'yyg', 'yyh', 'yyi', 'yyk', 'yyl', 'yym', 'yy n', 'yyp', 'yyq', 'yyr', 'yy s', 'yyt', 'yyv', 'yyw', 'yyx', 'yyy']
```

- range(4,4)長度產生範例:

```
['yyya', 'yyyc', 'yyyd', 'yyye', 'yyyf', 'yyyg', 'yyyh', 'yyyi', 'yyyk', 'yyyl', 'yyym', 'yyyn', 'yyyp', 'yyyq', 'yyyr', 'yyys', 'yyyt', 'yyyv', 'yyvw', 'yyyy']
```

- range(5,5)長度產生範例:

```
['yyytq', 'yyvvd', 'yyvvg', 'yyv vk', 'yyvvl', 'yyvvn', 'yyvvt', 'yyvvv', 'yyvwy', 'yyvyd', 'yyvyg', 'yyvyl', 'yyvyn', 'yyvyp', 'yyv yq', 'yyv yr', 'yyv ys', 'yyv yt', 'yyv yv', 'yyv yw', 'yyv yx', 'yyv yy']
```

- range(6,6)長度產生範例:

```
['yyvvtl', 'yyvvvs', 'yyv wyl', 'yyv ydf', 'yyv yg l', 'yyv ygm', 'yyv yhr', 'yyv yll', 'yyv yma', 'yyv ymd', 'yyv ysa', 'yyv ylt', 'yyv ydfy', 'yyv ygld', 'yyv ygmd', 'yyv yhre', 'yyv ylle', 'yyv ymav', 'yyv ymdv', 'yyv ypft', 'yyv yvsa', 'yyv ywylt', 'yyv ydfy', 'yyv ygld', 'yyv ygmd', 'yyv yhre', 'yyv ylle', 'yyv ymav', 'yyv ymdv', 'yyv ypft']
```

- range(7,7)長度產生範例:

```
['yyvvtl', 'yyvvvs', 'yyv wyl', 'yyv ydf', 'yyv yg l', 'yyv ygm', 'yyv yhr', 'yyv yll', 'yyv yma', 'yyv ymd', 'yyv ysa', 'yyv ylt', 'yyv ydfy', 'yyv ygld', 'yyv ygmd', 'yyv yhre', 'yyv ylle', 'yyv ymav', 'yyv ymdv', 'yyv ypft', 'yyv yvsa', 'yyv ywylt', 'yyv ydfy', 'yyv ygld', 'yyv ygmd', 'yyv yhre', 'yyv ylle', 'yyv ymav', 'yyv ymdv', 'yyv ypft']
```

不同裁切長度在Naive Bayes Model所得到的模型分類準確度: (切成range(5,5)的準確度是最高的97.2%)

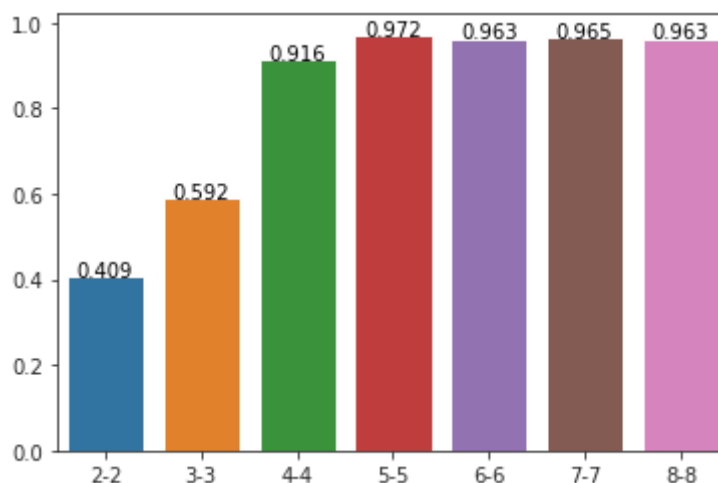


圖3.3 不同n-gram裁切長度的模型準確率

b. 特徵抽取: 使用tokenizer取特徵

首先針對sequence字串的長度統計:

```
1 seqs_length = []
2 for i in range(len(seqs)):
3     seqs_length.append(len(seqs[i]))
4
5 np.max(seqs_length), np.min(seqs_length), round(np.mean(seqs_length)), round(np.std(seqs_length)), len(seqs_length)

(5070, 2, 278, 289, 197371)
```

我們知道長度最長為5070, 最短為2, 平均長度為278, 標準差為289, 全部資料筆數為197371。在考量深度學習模型使用的GPU記憶體大小有限(只有一張顯示卡), 故我們需將sequence使用(keras)tokenizer轉換時的最大長度設定為平均長度的一倍標準差, 以便模型訓練。

```
max_length = int(round(np.mean(seqs_length)) + (1*round(np.std(seqs_length))))
```

關於資料集維度的問題, 使用n-gram(以range(4,4)為例), 在n-gram處理後維度大小來到169600的長度(如下圖示), 而且是一個稀疏矩陣, 這樣的特徵處理較不利於Tree-based model的模型學習, 在後續實作結果的比較我們可以明顯看到採用n-gram前處理的Tree-based model模型成效都低於採用機率基礎的模型。

```
[ ] 1 #原本資料集的維度
    2 X_train.shape, X_test.shape, y_train.shape, y_test.shape

((157896,), (39475,), (157896,), (39475,))

[ ] 1 #切完n-gram.range(4,4)後的維度: 欄位長度來到169600
    2 type(X_train_df), X_train_df.getformat, X_train_df.shape, X_test_df.shape

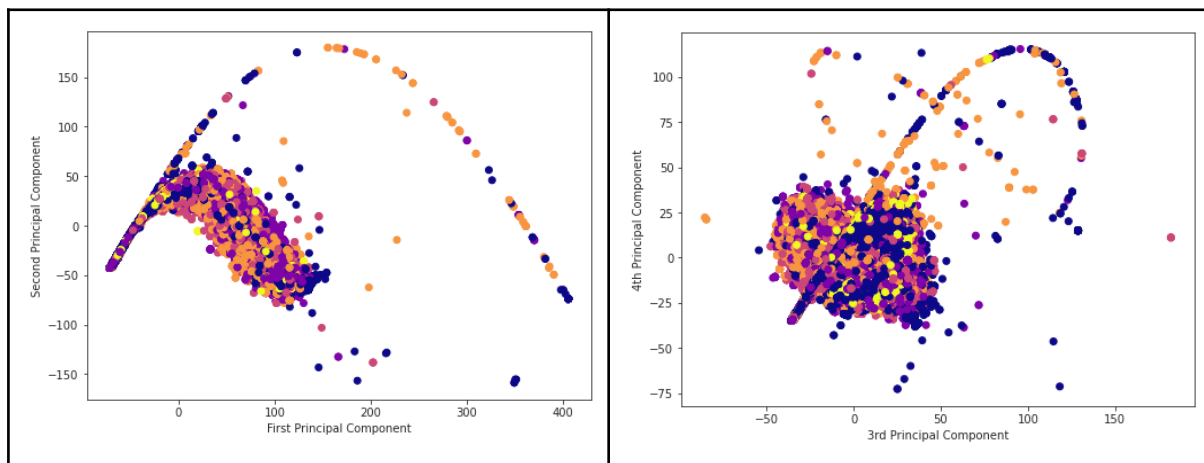
(scipy.sparse.csr.csr_matrix,
 <bound method spmatrix.getformat of <157896x169600 sparse matrix of type '<class 'numpy.int64''>'
   with 40021591 stored elements in Compressed Sparse Row format>>,
 (157896, 169600),
 (39475, 169600))
```

採用tokenizer取特徵的方法, 在我們的實作中是使用平均sequence長度取一倍標準差為固定的維度大小, 計算得到567維度長度來表示每一個sequence的特徵長度。這樣的特徵工程方法是將每個sequence位置, 視為一個特徵(維度/欄位), 位置前後之間在生物化學原理上是有關聯的, 並非各自獨立。(這部份在後續使用PCA線性降維得不到更好的結果可以驗證)

我們使用XGBoost (Tree-based model)搭配tokenizer取567維度長度, 得到約86%的分類準確度, 分類模型報告如下圖示:

	precision	recall	f1-score	support
RIBOSOME	0.91	0.96	0.93	12136
HYDROLASE	0.78	0.86	0.82	9528
TRANSFERASE	0.83	0.77	0.80	7698
OXIDOREDUCTASE	0.91	0.80	0.85	6929
IMMUNE SYSTEM	0.90	0.82	0.86	3184
accuracy			0.86	39475
macro avg	0.87	0.84	0.85	39475
weighted avg	0.86	0.86	0.86	39475

後續我們嘗試使用PCA來將567維度進行降維, 效果並沒有明顯提昇, 可以得知sequence資料的各個位置, 並不適合線性獨立的特徵處理方法。進階我們可以改採深度學習的embedding方法, 來得到非線性獨立的特徵處理。下圖是PCA第一 vs 第二, 第三 vs 第四個Component所繪製的散佈圖, 可以明顯看出各類別資料點並沒有有效的被區隔開來。



3.4 模型

在本小節我們將使用多個機器學習(ML)與深度學習(DL)模型來進行模型訓練及評估，以便比較各別模型的成效以及ML模型與DL模型在這個資料集的成效表現上是否有差異。

3.4.1 ML模型

以下呈現機器學習模型訓練集的模型成效報表(以各模型逐一表列):

Naive Bayes Model (MultinomialNB) – 做為 **baseline model**

初步ML模型分類預測準確率約92%

	precision	recall	f1-score	support
RIBOSOME	0.87	0.93	0.90	9501
HYDROLASE	0.98	0.84	0.90	3261
TRANSFERASE	0.96	0.91	0.93	7023
OXIDOREDUCTASE	0.98	0.91	0.94	12027
IMMUNE SYSTEM	0.83	0.94	0.88	7663
accuracy			0.92	39475
macro avg	0.92	0.91	0.91	39475
weighted avg	0.92	0.92	0.92	39475

RandomForest (隨機森林) - 預測準確率約32%

	precision	recall	f1-score	support
HYDROLASE	0.00	0.00	0.00	9501
IMMUNE SYSTEM	0.99	0.22	0.36	3261
OXIDOREDUCTASE	0.00	0.00	0.00	7023
RIBOSOME	0.31	1.00	0.47	12027
TRANSFERASE	0.00	0.00	0.00	7663
accuracy			0.32	39475
macro avg	0.26	0.24	0.17	39475
weighted avg	0.18	0.32	0.17	39475

Adaptive Boosting (Adaboost) - 預測準確率約60%

	precision	recall	f1-score	support
HYDROLASE	0.39	0.77	0.52	9501
IMMUNE SYSTEM	0.94	0.64	0.76	3261
OXIDOREDUCTASE	0.73	0.10	0.18	7023
RIBOSOME	0.93	0.91	0.92	12027
TRANSFERASE	0.44	0.34	0.38	7663
accuracy			0.60	39475
macro avg	0.69	0.55	0.55	39475
weighted avg	0.67	0.60	0.57	39475

Support Vector Machine (SVM) - 預測準確率約72%

	precision	recall	f1-score	support
HYDROLASE	0.54	0.73	0.62	9501
IMMUNE SYSTEM	0.86	0.90	0.88	3261
OXIDOREDUCTASE	0.66	0.58	0.62	7023
RIBOSOME	0.97	0.89	0.93	12027
TRANSFERASE	0.65	0.52	0.58	7663
accuracy			0.72	39475
macro avg	0.74	0.72	0.73	39475
weighted avg	0.74	0.72	0.73	39475

3.4.2 DL模型

以下呈現深度學習模型訓練集的模型成效及模型結構(以各模型逐一表列):

1D-CNN 模型 - 預測準確率約97.06%

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 567, 30)	780
conv1d (Conv1D)	(None, 567, 64)	11584
max_pooling1d (MaxPooling1D)	(None, 283, 64)	0
conv1d_1 (Conv1D)	(None, 283, 32)	6176
max_pooling1d_1 (MaxPooling1D)	(None, 141, 32)	0
flatten (Flatten)	(None, 4512)	0
dense (Dense)	(None, 128)	577664
dense_1 (Dense)	(None, 5)	645
Total params: 596,849		
Trainable params: 596,849		
Non-trainable params: 0		

	precision	recall	f1-score	support
HYDROLASE	0.96	0.96	0.96	9594
IMMUNE SYSTEM	0.96	0.95	0.96	3164
OXIDOREDUCTASE	0.98	0.97	0.97	7087
RIBOSOME	0.99	1.00	0.99	12114
TRANSFERASE	0.96	0.96	0.96	7516
accuracy			0.97	39475
macro avg	0.97	0.97	0.97	39475
weighted avg	0.97	0.97	0.97	39475

LSTM 模型 - 預測準確率約97.75%

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, None, 567)	17010
lstm (LSTM)	(None, 256)	843776
dense (Dense)	(None, 5)	1285
Total params: 862,071		
Trainable params: 862,071		
Non-trainable params: 0		

	precision	recall	f1-score	support
HYDROLASE	0.97	0.97	0.97	9542
IMMUNE SYSTEM	0.96	0.97	0.97	3179
OXIDOREDUCTASE	0.97	0.98	0.97	7032
RIBOSOME	0.99	0.99	0.99	12118
TRANSFERASE	0.97	0.95	0.96	7604
accuracy			0.98	39475
macro avg	0.97	0.97	0.97	39475
weighted avg	0.98	0.98	0.98	39475

Transformer 模型 - 預測準確率約92.18%

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 567)]	0
token_and_position_embedding (TokenAndPositionEmbedding)	(None, 567, 32)	19104
transformer_block (TransformerBlock)	(None, 567, 32)	23232
global_average_pooling1d (GlobalAveragePooling1D)	(None, 32)	0
dropout_2 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 20)	660
dropout_3 (Dropout)	(None, 20)	0
dense_3 (Dense)	(None, 5)	105
=====		
Total params: 43,101		
Trainable params: 43,101		
Non-trainable params: 0		

	precision	recall	f1-score	support
HYDROLASE	0.90	0.88	0.89	9529
IMMUNE SYSTEM	0.92	0.89	0.90	3191
OXIDOREDUCTASE	0.95	0.89	0.92	6976
RIBOSOME	0.98	0.99	0.98	12152
TRANSFERASE	0.84	0.92	0.88	7627
accuracy			0.92	39475
macro avg	0.92	0.91	0.91	39475
weighted avg	0.92	0.92	0.92	39475

四、實作結果

本章節我們彙整專題使用的所有機器學習與深度學習模型在訓練集與測試集的模型成效，做成果比較。同時也運用混淆矩陣來顯示出模型在測試集的預測結果分佈狀況，逐一呈現如下圖表。

- 模型成果比較：

表4.1 模型成果比較表

模型類型	模型名稱	Train-accuracy	Test-accuracy
ML Model	Naive Bayes	93.00%	91.59%
	RandomForest	32.60%	32.27%
	Adaboost	60.22%	59.92%
	SVM	73.43%	72.49%
	XGBoost	84.82%	83.18%
DL Model	1D-CNN	99.28%	97.06%
	LSTM	<u>99.45%</u>	<u>97.75%</u>
	Transformer	94.18%	92.18%

- 混淆矩陣(Confusion matrix): 在測試集的預測成效比較。

Class List: 各類別編號代表的名稱

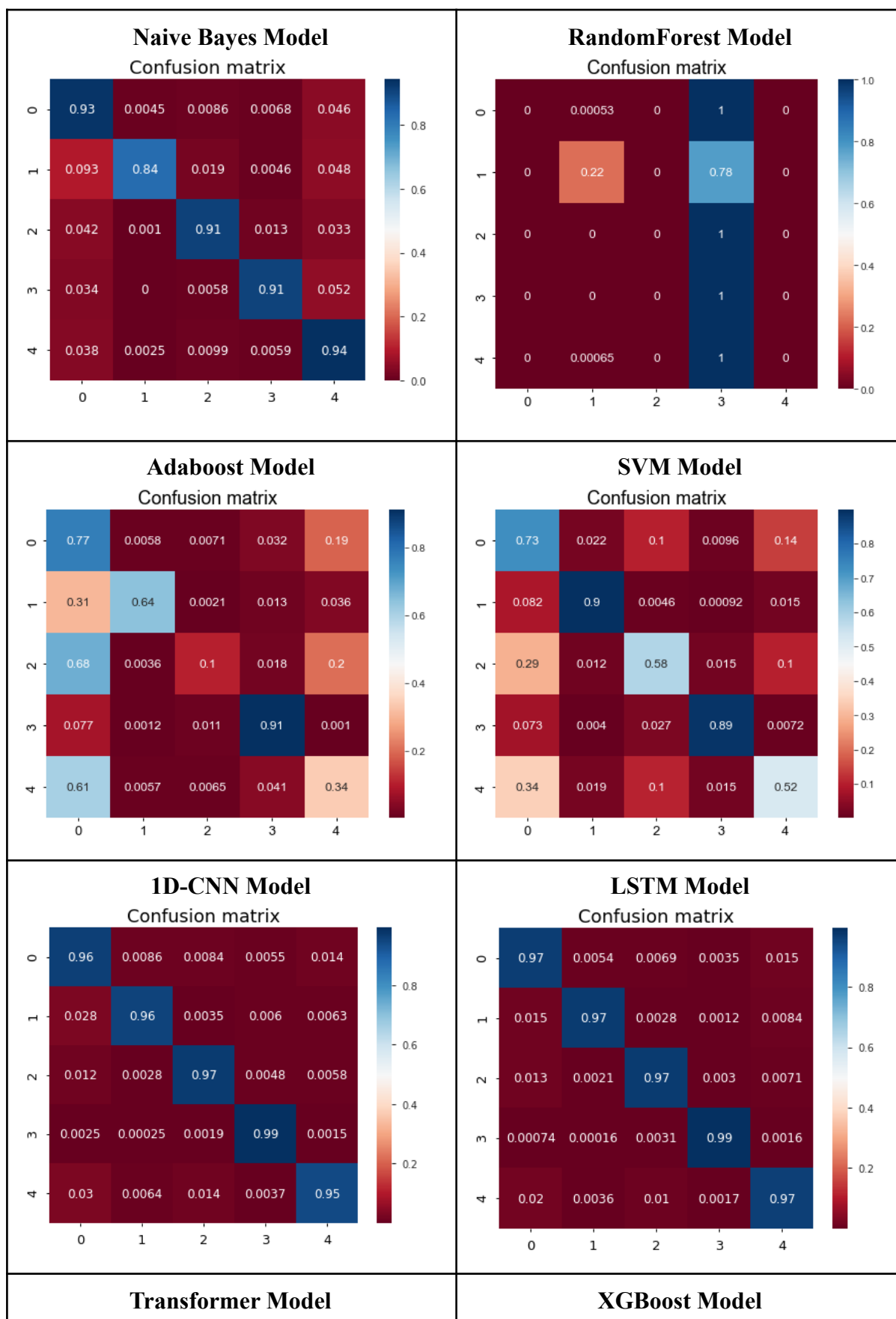
0:核糖蛋白(Ribosome)

1:水解酶(Hydrolase)

2:轉移酶(Transferase)

3:氧化還原酶(Oxidoreductase)

4:免疫系統(Immune system)



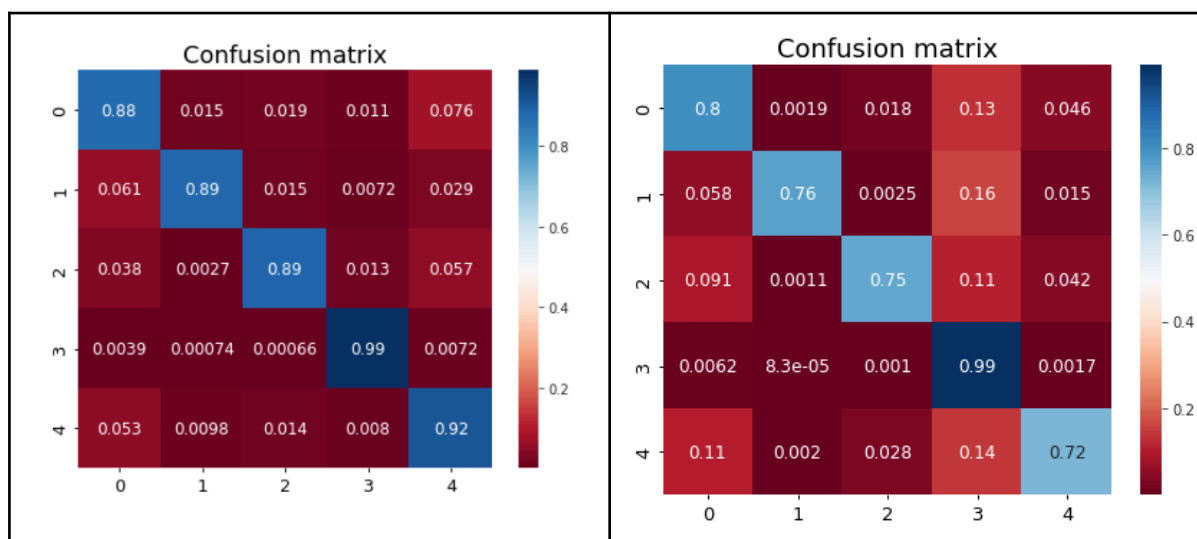


圖4.1 混淆矩陣(使用測試集)

五、結論

目前專題使用了八種模型做分類建模測試，我們可以發現在這個蛋白質氨基酸序列分類任務上，深度學習模型的效果良好，我們並沒有很細節的去調整及優化參數，只用幾個基本的模型架構(LSTM, 1D-CNN)即可達到97%的分類準確率，使用較複雜的模型如Transformer的預測成效並沒有很顯著優於LSTM模型(也有可能是訓練參數沒有設定好，造成的結果)，但普遍來看深度學習模型的成效比機器學習模型較優。

蛋白質結構分為四級，本專題目前只針對第一級結構做概念驗證，以瞭解深度學習模型在這類問題的建模成效良好，這也包括運用NLP的特徵抽取，例如：n-gram、tokenizer、embedding、等技巧，來取得有效的模型特徵。

因實作時間上沒有來得及完整的做BERT建模測試，但多數的論文都有談到BERT模型的非監督式學習方式，很適合這類難以大量標記的建模任務。是我持續進行此類研究的一個方向，希望除了一級結構的建模，也能進一步進行二級及三級結構的建模實作。

參考文獻

- [1] kaggle - protein DNA sequence dataset, <https://www.kaggle.com/shahir/protein-data-set>
- [2] Original data set down loaded from <http://www.rcsb.org/pdb/>
- [3] 機器學習遇見生物學:詳解蛋白質摺疊預測中的算法,
<https://zhuanlan.zhihu.com/p/98960312>
- [4] 蛋白質摺疊(英語:Protein folding)是蛋白質獲得其功能性結構和構象的物理過程。
<https://zh.wikipedia.org/wiki/%E8%9B%8B%E7%99%BD%E8%B4%A8%E6%8A%98%E5%8F%A0>
- [5] 蛋白質一級結構
<https://zh.wikipedia.org/wiki/%E8%9B%8B%E7%99%BD%E8%B3%AA%E4%B8%80%E7%B4%9A%E7%B5%90%E6%A7%8B>
- [6] 蛋白質氨基酸序列表示示意圖, <https://bair.berkeley.edu/blog/2019/11/04/proteins/>
- [7] 可解釋 AI (XAI) 系列 — SHAP <https://medium.com/ai-academy-taiwan/2c600b4bdc9e>

附錄

[1] 第一級結構預測參考實作



[colab]本專題實作程式檔案連結:

N-gram ML :

https://drive.google.com/file/d/1LkRzmC72LhKe22b6KNPT2dbIPCmbllhLx/view?usp=share_link

1DCNN :

https://drive.google.com/file/d/1yOXeYdiEWb_o7HnvSbGt0mQsm1XpQSOi/view?usp=share_link

LSTM :

https://drive.google.com/file/d/1gS14Jv8HlxVdI77YyWQTV-DEjx3XzT-J/view?usp=share_link

Transformer :

https://colab.research.google.com/drive/12ANMgciU0JUBTBMz8Lf8cJ5X0ZDf7_w-?usp=share_link

RandomForest :

https://drive.google.com/file/d/1SQu8cvVjp45NxO9baKLJ0ox-Z8nEvwf2/view?usp=share_link

Adaptive Boosting (Adaboost):

https://drive.google.com/file/d/1B81VuhIHgQPRa2kb71p0GCbHyJNdi91M/view?usp=share_link

Support Vector Machine (SVM):

https://drive.google.com/file/d/1gOECfapzTBscq8VaCYAy4F-qEPUuTwO_/view?usp=share_link

XGBoost:

https://drive.google.com/file/d/1SZs7nKVwCU1hCay859f8zSvXY4AkLnIx/view?usp=share_link

[2] 第二級結構預測參考實作

[pytorch] Tasks Assessing Protein Embeddings (TAPE):

<https://github.com/songlab-cal/tape>

This is the code associated with our original paper and benchmark.(2019)

<https://github.com/songlab-cal/tape-neurips2019>

Preprint is available at <https://arxiv.org/abs/1906.08230>

[paperwithcode] TAPE

<https://paperswithcode.com/paper/evaluating-protein-transfer-learning-with>

ProteinBERT: A universal deep-learning model of protein sequence and function (2021)

<https://www.biorxiv.org/content/10.1101/2021.05.24.445464v1>

[github] ProteinBERT

https://github.com/nadavbra/protein_bert

[3] 第三級結構預測參考實作

Highly accurate protein structure prediction with AlphaFold (2021)

<https://www.nature.com/articles/s41586-021-03819-2>

[github] AlphaFold (deepmind)

<https://github.com/deepmind/alphafold>

[github] ColabFold (*有方便的操作介面)

<https://github.com/sokrypton/ColabFold>

[Term Project]

請自行找一個有興趣分析的資料集, 或使用之前作業之資料集, 並使用Scikit-Learn、深度學習模型(可用Keras或Tensorflow...等)完成本次學期專題, 選擇三個以上模型去作比較分析

請將期末報告實作之內容撰寫一份報告書

內容須包含：題目、動機、資料集敘述、分析工具、實作與評估方法、流程(分析流程圖、結果截圖等)、分析結果與結論(或其他補充內容)

E3作業繳交: 報告書(pdf檔) + 程式碼 (將以上檔案壓縮成 組別_題目.zip, 一組繳交一份即可)

繳交期限: 1/9 (一) 23:59

題目: 結構蛋白質氨基酸序列分類預測

摘要: (1p) - 林德全

關鍵字:

一.前言/動機 (1p) - 林德全

二.實作方法 (4p) - 徐嘉

三.實作步驟 (10p) - 許秀琳 / 林德全

四.實作結果 (2p) - 許秀琪 / 林德全

五.結論 (1p) - 許秀琪

參考文獻 (1p)

字形: Times New Roman

標題: 20, 粗體

內文: 12