Computation in Datascience - PartII - Final Project

B06702064 會計五 林聖硯

資料描述

此次資料包含4個檔案，有以下兩個任務。

- GDSC_TCGA_Docetaxel.csv, Patients_TCGA_Docetaxel.csv: 用來預測16個patients的藥物二元分類問題(resistant, sensitive)
- GDSC_PDX_Gemcitabine.csv, Patients_PDX_Gemcitabine.csv: 預測25個patients的藥物二元分類問題，並以25個patients的labels驗證模型的準確率

由於GDSC_TCGA_Docetaxel.csv以及GDSC_PDX_Gemcitabine.csv兩個檔案中並沒有我們要預測的label，我們必須透過IC50之轉換才能得到最後的label(S/R)。

- 在GDSC_TCGA_Docetaxel.csv中，最大用藥濃度為0.0125，若IC50大於最大用藥濃度則此資料之label為R
- 在GDSC_PDX_Gemcitabine.csv中，最大用藥濃度為1.024，若IC50大於最大用藥濃度則此資料之label為R

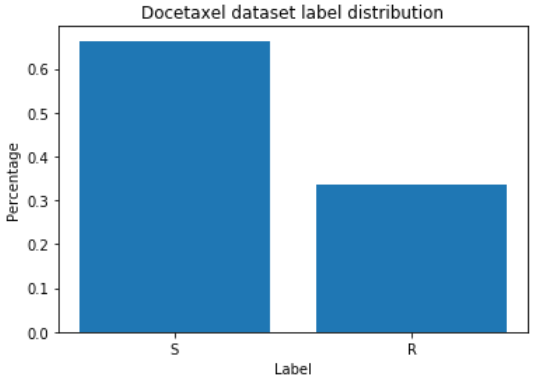
Exploratory data analysis(EDA)

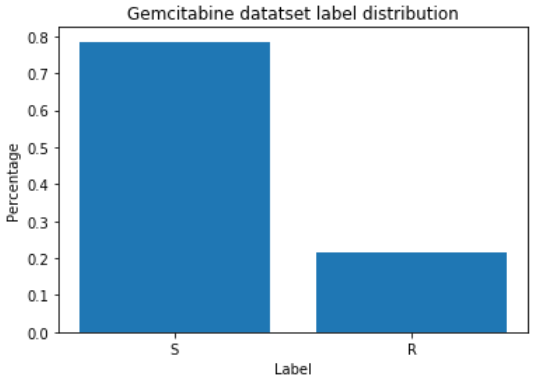
在進行資料處理以及建模前，我先對資料進行EDA分析，藉以找出重要資訊協助後續資料處理，而我在進行EDA時發現了以下幾件重要的事情。

1. 兩個資料集之資料筆數都遠小於features的數量。去除 `IC50` 、 `gene_expression` 以及 `label` 之欄位後，Docetaxel資料集之特徵值為**16,170個欄位**，遠遠大於**850筆觀測值**，而Gemcitabine資料集之特徵值為**16,192個欄位**，遠遠大於**866筆觀測值**。因為$Rank(X) << \#$ of

features，這會使得後續模型建模時容易出現overfit、或是導致regression失效等狀況，故特徵篩選會是此任務中最重要的一個步驟。

2. 觀察兩個資料集內label的分布。在Docetaxel資料集內，S與R比率分別為66.353%以及33.647%；而在，在Gemcitabine資料集內，S與R比率分別為78.522%以及21.478%。兩個資料集內都有資料不平均(imbalanced data)的狀況。故後續的任務中，我都有使用**repeated stratified K-Fold**來進行特徵篩選與建立模型，能夠使這些過程的結果較為robust。





特徵篩選

1. Filter Method - Anova F-test
   雖然上課教的anova f-test是categorical features對上categorical label的狀況，但其實anova f-test也可以是numerical features對上categorical label的情況。而在兩個資料集中，我使用anova f-test選出來F-score前30高的features(因為$30 \approx \sqrt{850}$)，他們分別為：

Docetaxel資料集之Anova features

['ABCB1', 'GPX8', 'YAP1', 'IGF2BP2', 'MYRIP', 'IGFBP6', 'RAB31', 'EXT2', 'ZNF124', 'BAG3', 'GFPT2', 'COL7A1', 'SEC11C', 'PACSIN3', 'FAM20C', 'CAV1', 'AXL', 'BCAR3', 'PRNP', 'RRAS', 'RAB34', 'CAV2', 'ATP8A1', 'UCP2', 'EMP1', 'HEG1', 'PTGR1', 'RAB11FIP5', 'PDLIM4', 'PARVA']

Gemcitabine資料集之Anova features

['CCT4', 'SLFN11', 'HDAC11', 'ANP32B', 'MTHFD1L', 'DIMT1', 'NPM3', 'TFF3', 'DYNLRB2', 'SYT7', 'ATP6AP1', 'PPWD1', 'PRR15L', 'MGP', 'PPP1R3D', 'NOB1', 'MB', 'BTAF1', 'MAT2B', 'SNHG1', 'RPS12', 'RPL11', 'PBX1', 'PIP', 'LYAR', 'IGFBP5', 'GUK1', 'USP45', 'TAF1D', 'SNX5']

另外，我也挑選了兩種自帶特徵重要程度(feature importance)的演算法，兩種徵篩選方法都有經過**10-Repeated Stratified 5-Fold Cross Validation**使得篩選結果更加穩定。

2. Embedded Method - Logistic regression with L1 regularization

   Logistic regression是一種常用的、能解決二元分類問題的演算法，而加上 L1 regularization後，如果是不重要的features，模型很容易讓這個features的coefficient強制設為0，也因此能透過這樣子的特性來篩選重要的變數。在透過10-Repeated Stratified 5-Fold Cross Validation以及使用Accuracy當作模型篩選標準後，在第一個Docetaxel資料集中，coefficient不為0的特徵分別有**123個**；但在第二個Gemcitabine資料集中，L1 logistic regression所有的features之係數均為0，故沒有辦法使用此方法挑出features，推測是特徵數太多導致模型無法收斂至最佳解所導致($Rank(X) << $ # of features容易造成regression在推估參數時出現問題)。

Docetaxel資料集之L1 Reg features

['ADARB1', 'ADCY3', 'APIP', 'APOC1', 'APTX', 'ASMT', 'AWAT2', 'BID', 'C22orf42', 'C3orf67', 'C5', 'CCDC158', 'CCNC', 'CCT4', 'CD300LD', 'CDCA7', 'CDK5RAP2', 'CEP78', 'CHCHD4', 'CISD1', 'CNGB1', 'CNN2', 'COL27A1', 'COL7A1',

'COPS2', 'CSNK1E', 'DDIT4L', 'DGKK', 'DUSP9', 'EIF3A', 'ELP4', 'ETF1', 'EXT2', 'F10', 'FAHD2A', 'FAM20C', 'FBLN1', 'FBXO30', 'FKBP15', 'FOXL2', 'GABRA4', 'GART', 'GFM2', 'GFPT2', 'GGCT', 'GLI1', 'GPR153', 'GRHPR', 'GSDMD', 'HACL1', 'HDAC1', 'HEG1', 'HIBADH', 'HLA.DPB2', 'HSP90AB1', 'IGF2BP2', 'IGFBP6', 'IL11', 'INSL3', 'INVS', 'IPO7', 'ITPRIP', 'KCNJ1', 'KIF18A', 'KREMEN2', 'LIPG', 'LLGL1', 'MAFF', 'MDFI', 'MEDAG', 'MFAP2', 'MFF', 'MROH7', 'MRPS24', 'NAA50', 'NDUFAF7', 'NEIL1', 'NPM1', 'NQO2', 'NTNG2', 'OMG', 'ORC2', 'PACSIN3', 'PALM3', 'PDHX', 'PDLIM4', 'PHGDH', 'PITX1', 'PLD2', 'PMS1', 'PPP1CB', 'PRTG', 'RAB34', 'RASA2', 'REP15', 'RGP1', 'RNF8', 'RPAP1', 'RPP25L', 'SAAL1', 'SAR1A', 'SEMA4A', 'SERPINB3', 'SGK1', 'SH3BP5', 'SIM2', 'SLC15A2', 'SLC29A4', 'SMARCA1', 'SMU1', 'SNX3', 'SOCS2', 'STOML2', 'SUPT7L', 'SYCP2L', 'TKTL2', 'TMEM184B', 'TSNAXIP1', 'UBAP1', 'URB1', 'ZNF558', 'ZNF717', 'ZNF846']

Gemcitabine資料集之L1 Reg features

[] (無)

3. Embedded Method - Random Forest Classifier
   我最後一個特徵篩選的方法與2.類似，也是一個embedded method，只是改成隨機森林(random forest)演算法。

隨機森林是一種ensembling learning中的**bagging模型**。演算法每次會從樣本中抽樣(bootstrap)，並從features裡面選出$p = \sqrt{m}$個features出來訓練decision tree，重複上述的結果多次(次數為可調整的參數)，在最後透過majority vote的方式決定分類別的結果。Bagging的優點在於原始訓練樣本中有噪聲資料(不好的資料)，透過Bagging抽樣就有機會不讓有噪聲資料被訓練到，所以可以降低模型的不穩定性。
而這個演算法中最重要的就是他能夠計算每一個特徵在每次分割時能得到最大的資訊增益(information gain)，並且選擇用哪個特徵來分割。也因此能夠良好劃分label的特徵得到的資訊增益越高，最終模型出來的feature importance也會越

高，所以我們可以拿random forest的feature importance來當作我們篩選特徵的標準。最後，在第一個Docetaxel資料集中，我觀察到feature importance的斷點大約落在0.001附近，高於於0.001的特徵比較少，而低過0.001的特徵非常多。故我用0.001當作threshold來當作篩選標準，篩選出以下24個feature。而在第二個Gemcitabine資料集中，並沒有特別的斷點，故與1.方法相同，選出前30個最高feature importance的features。

Docetaxel資料集之Ranfom forest features

['ABCB1', 'GFPT2', 'IGFBP6', 'PACSIN3', 'COL7A1', 'EXT2', 'BCL2', 'DDC', 'YAP1', 'GPX8', 'PLS3', 'SERPINE1', 'GGCX', 'COG2', 'PTGR1', 'ITPRIP', 'ADAMTSL2', 'PTPRO', 'WWTR1', 'OR51E1', 'MYRIP', 'IGF2BP2', 'TPM2', 'NOL4']

Gemcitabine資料集之Ranfom forest features

['CCT4', 'CLPX', 'TFF3', 'SLFN11', 'VAPB', 'ZNF552', 'TCTA', 'TFB1M', 'NACA', 'ZNF121', 'CCNB1IP1', 'COL10A1', 'PPDPF', 'GRHL1', 'ALDH3B2', 'SLC9A7', 'RCC1', 'CDH1', 'NELFCD', 'DIMT1', 'ZBTB7C', 'ANP32B', 'HS3ST6', 'GUK1', 'CPQ', 'VEGFB', 'PANK2', 'NMI', 'MAGED2', 'CERS6']

資料前處理

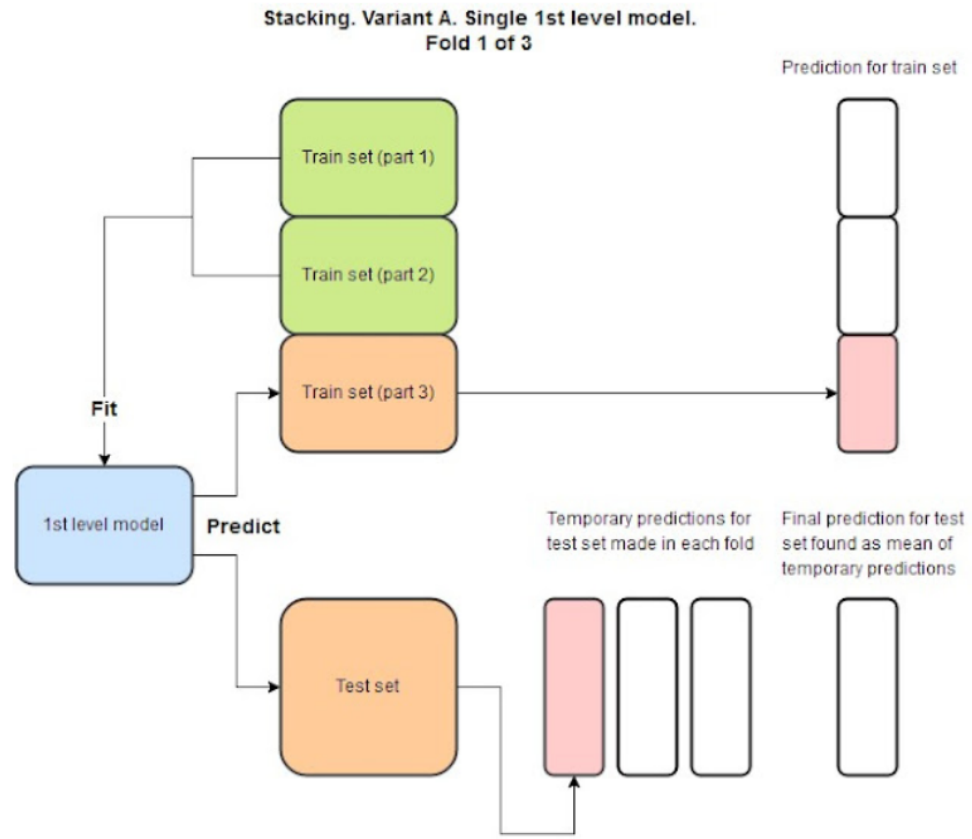
1. 在最後的特徵處理中，我聯集三個方法得到的features做為最後資料集的特徵
2. 為了讓模型收斂加速、並增加收斂到良好狀態的機率，我對這X個特徵分別進行normalization

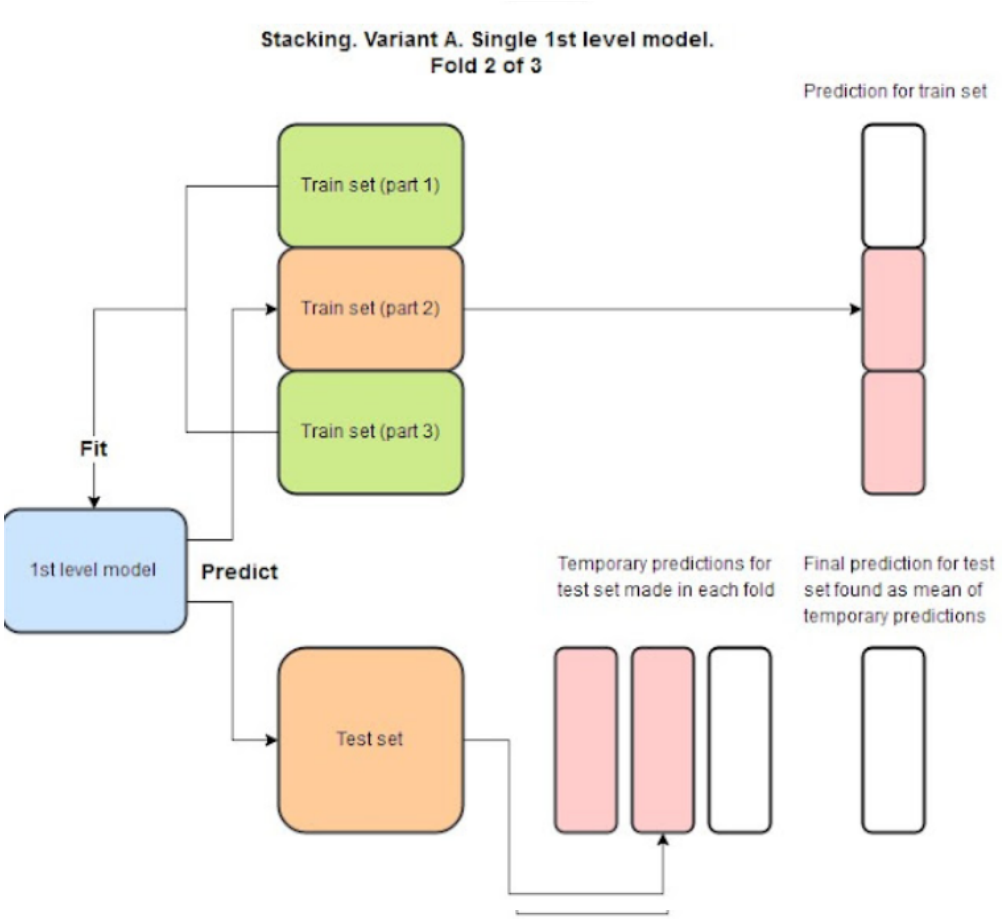
建模

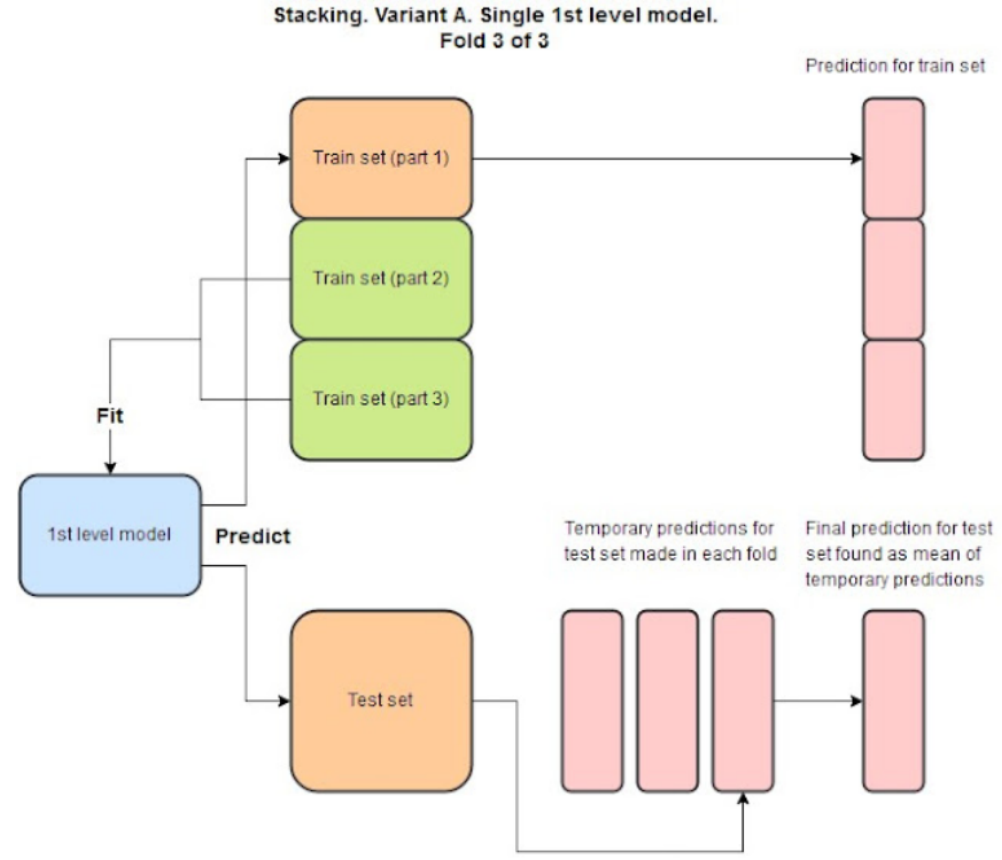
在建模的部分，我使用了一個名為**stacking**的機器學習技巧。Stacking中文稱為堆疊法，首先產生出m個 base learners(模型)彼此間並互相無關連，例如第一個 learner 為 KNN 第二個為決策樹。訓練完 m 個模型後，我們要把這m個模型合併在一起。合併的方式是我們另外再訓練一個模型，這個模型把m個base learner的輸出當成新的模型的輸入

因此我們會根據這 m 個特徵利用集成式學習其中的演算法來學習一個模型並預測最終結果。最後的 Ensemble model通常使用一個簡單的logisitic regression或是淺層的Neuron Netwrok來做學習，Ensemble model 的目的是把每一個 base learners 的輸出當成線索，並把這些線索想辦法做整合來得到最終的答案。Stacking也是一個在Kaggle常用的競賽技巧，幾乎比較前面的名次都一定會做Stacking來衝高Accuracy。

Stacking的概念如下圖所示：







在這裡我刻意選擇了以下四個演算法相差較遠的base

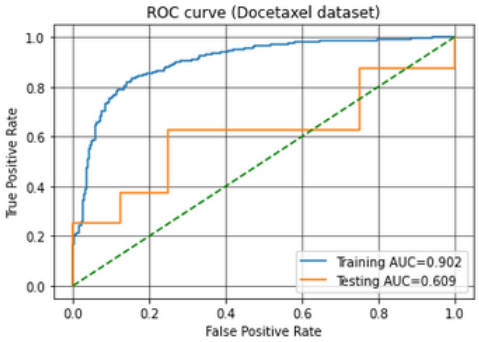
learners，讓每個模型能夠學到不一樣的線索，最後再透過logistic regression作為模型輸出。以下的每個模型都有透過10-Repeated Stratified 5-Fold Cross Validation找到最佳的超參數，預測出結果後再進行第二層(Ensemble model)的模型訓練。

1. Ridge Regression (線性模型)
2. Random Forest (能捕捉非線性關係的模型)
3. Support Vector Machine(將資料轉換至高維度後再用一個hyperplane進行分類之模型)
4. K-Nearest Neighbor(透過K個附近的資料點的label決定你屬於哪個class)

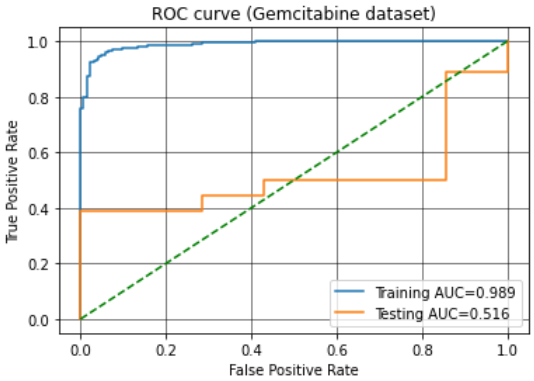
Testing資料集結果與結論

在透過以上的過程訓練完stacking classifier後，我們就能拿兩個任務中的testing資料集(也就是patient的資料)來預測，以下分別畫出兩個任務中的AUC以及ROC。

Docetaxel



Gemcitabine



從結果來看，Gemcitabine之任務難度遠大於Docetaxel，testing AUC為0.516，只比隨機猜測(AUC=0.5)的結果好一點

點，從前面的L1 logistic regression之結果就能略知一二，我認為可能的原因有兩個。

1. 此任務中training AUC有0.989，很有可能是我的features篩選錯誤所導致，使得模型overfit training dataset太嚴重(但我認為機率非常低，因為前面我所有的特徵篩選或建模過程都有使用stratified repeated cross validation)
2. testing dataset的資料過少、且代表性不足。相比於866個觀察值的training set，testing dataset裡面只有25個patient，且她們的label分布(R/S比例為50/50)不與training set相同，很明顯地此testing dataset沒有辦法有效反映所有patient的資料分布，故在testing階段結果很差是可以預期的。(正如同老師上課所說的"training&testing data not homogenized"的問題)

考慮到以上兩個原因，我又嘗試了不同features數量(從上面聯集的features中隨機抽出)搭配模型設置來計算testing dataset的AUC，並建立以下的AUC表格得到最後的結果。
(以下X代表聯集中沒有那麼多features可以抽)

任務 / 特徵數量	p=10	p=20	p=30	p=40	p=50
Docetaxel	0.583	0.587	0.590	0.597	0.603
Gemcitabine	0.498	0.507	0.510	0.516	0.540

任務 / 特徵數量	p=60	p=70	p=80	p=90	p=100
Docetaxel	0.617	0.630	0.623	0.612	0.609
Gemcitabine	0.524	X	X	X	X