# Computation in Datascience - PartII - HW2

## B06702064 會計五 林聖硯

### 資料描述

此次資料包含3個檔案，GDSC_PDX_Paclitaxel.csv、CCLE_PDX_Paclitaxel.csv以及GDSC_Paclitaxel_info.csv。

- GDSC_PDX_Paclitaxel_info存放我們要預測的類別，但原資料裡面沒有標註的欄位，故在這裡我自行進行標註。若 `IC50 < MAX_CONC_MICROMOLAR` ，則此資料的label為S，反之則為R。最後新增一個名為"label"的column來存放每筆資料的label
- GDSC_PDX_Paclitaxel.csv存放能夠預測以上label的features

### 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. 此資料集之觀測術以及feature數差距非常的大，去除 `MAX_CONC_MICROMOLAR` 、 `IC50` 以及 `CELL_LINE_NAME` 欄位後，特徵值為**16,190個欄位**，遠遠大於**399筆觀測值**。因為 $Rank(X) << \#\ \text{of features}$，這會使得後續模型建模時容易出現overfit的狀況，故特徵篩選會是此任務中最重要的一個步驟。
2. 觀察label的分布，S與R在資料中的比率分別為71.18%以及28.82%，有些微的資料不平均(imbalanced data)的狀況。後續資料處理以及建模的步驟中，若能加上oversampling以及undersampling的技巧可能可以使得模型的準確度進一步提升。

### 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。而我最後由anova f-test選出來F-score前20高的features，為['ABCB1', 'DOK4', 'SEZ6L2', 'PRPF4B', 'TBX3', 'RGS5', 'GPR22', 'COLEC11', 'SLC6A2', 'TFAP2B', 'ASTN2', 'C1QL4', 'PRIM2', 'PCSK1N', 'TRIM67', 'PNMA3', 'PODXL2', 'PHYHIPL', 'CRH', 'TMEM59L']。

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**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5-Fold Cross Validation以及使用Accuracy當作模型篩選標準後，最好的L1 Logistic regression中，coefficient不為0的特徵有**23個**，為['ANKFY1', 'C2orf68', 'CEP128', 'CHST2', 'GABPA', 'GYS2', 'HDAC1', 'ITGA4', 'JARID2', 'MDM1', 'OLFML2A', 'PPP3CB', 'PRIM2', 'PRPF4B', 'PTGES3', 'RAP2B', 'SMC6', 'TIMELESS', 'TRDMT1', 'TRIM25', 'TSLP', 'UBE2G1', 'ZNF318']。

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，我觀察到feature importance的斷點大約落在0.0015附近，高於於0.0015的特徵比較少，而超過0.0015的特徵非常多。故我用0.0015當作threshold來當作篩選標準，篩選出以下38個feature。

['ACTL7A', 'ALDH18A1', 'ABCB1', 'GYS2', 'KERA', 'TACR2', 'GPR22', 'THY1', 'FOXR1', 'C17orf64', 'MRPL14', 'NACC1', 'HSPA8', 'RBX1', 'MAP7D2', 'ATF1', 'PLCD4', 'DNAH10', 'CRH', 'HEATR4', 'ARHGDIA', 'RCC2', 'UCN3', 'FGF16', 'TMBIM6', 'PRRT4', 'SNCG', 'TRIM25', 'PRPSAP1', 'RECQL', 'TMEM203', 'VASP', 'PTOV1', 'ESPN', 'SLC7A11', 'LGALSL', 'IRAK1BP1', 'POLD2']

## 資料前處理

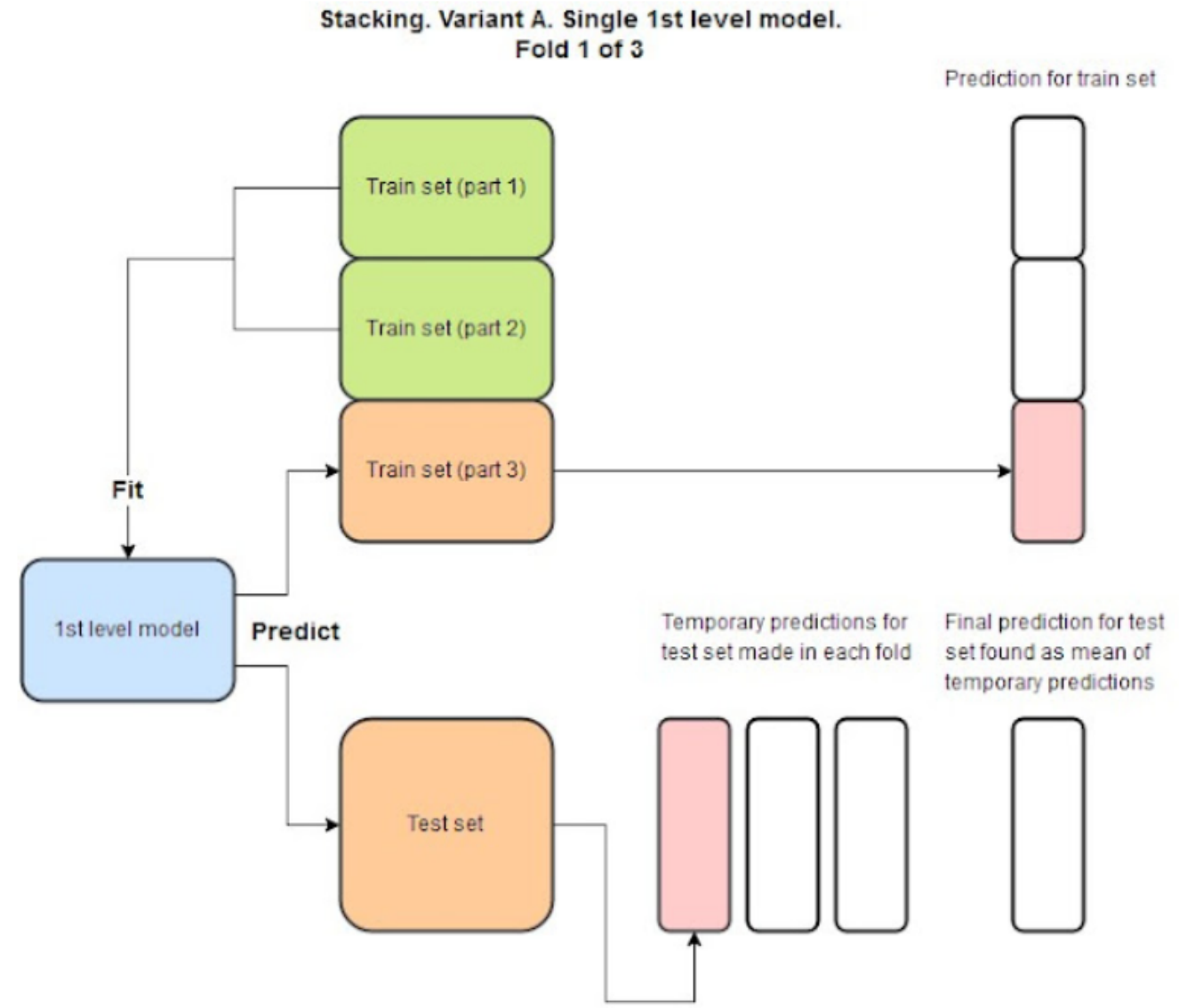
1. 透過聯集以上所得共**74個特徵**，並且移除兩個不在testing data set裡面的特徵，最後以**72個**特徵作為我最後的特徵數
2. 為了讓模型收斂加速、並增加收斂到良好狀態的機率，我對這72個特徵分別進行normalization

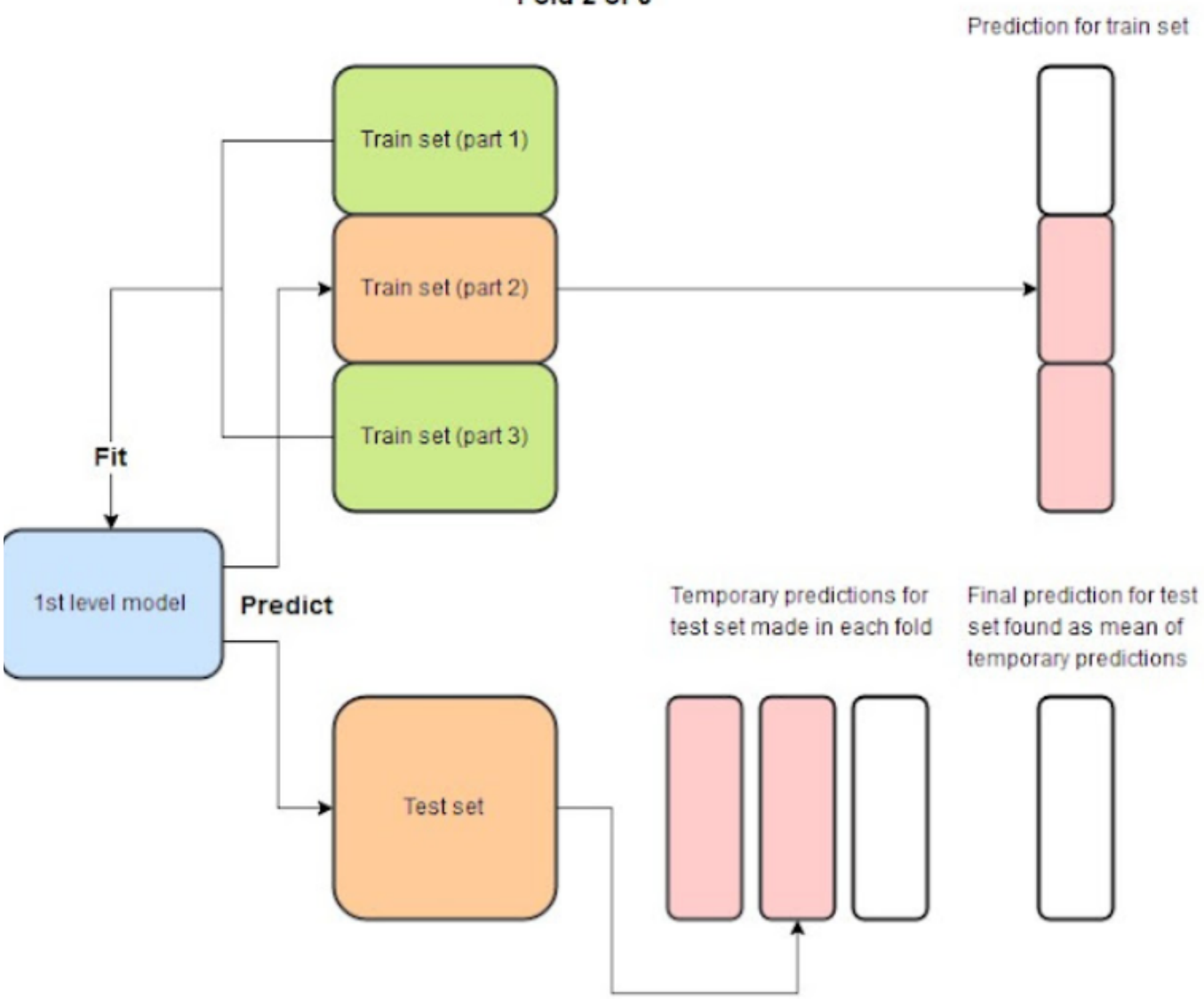
## 建模

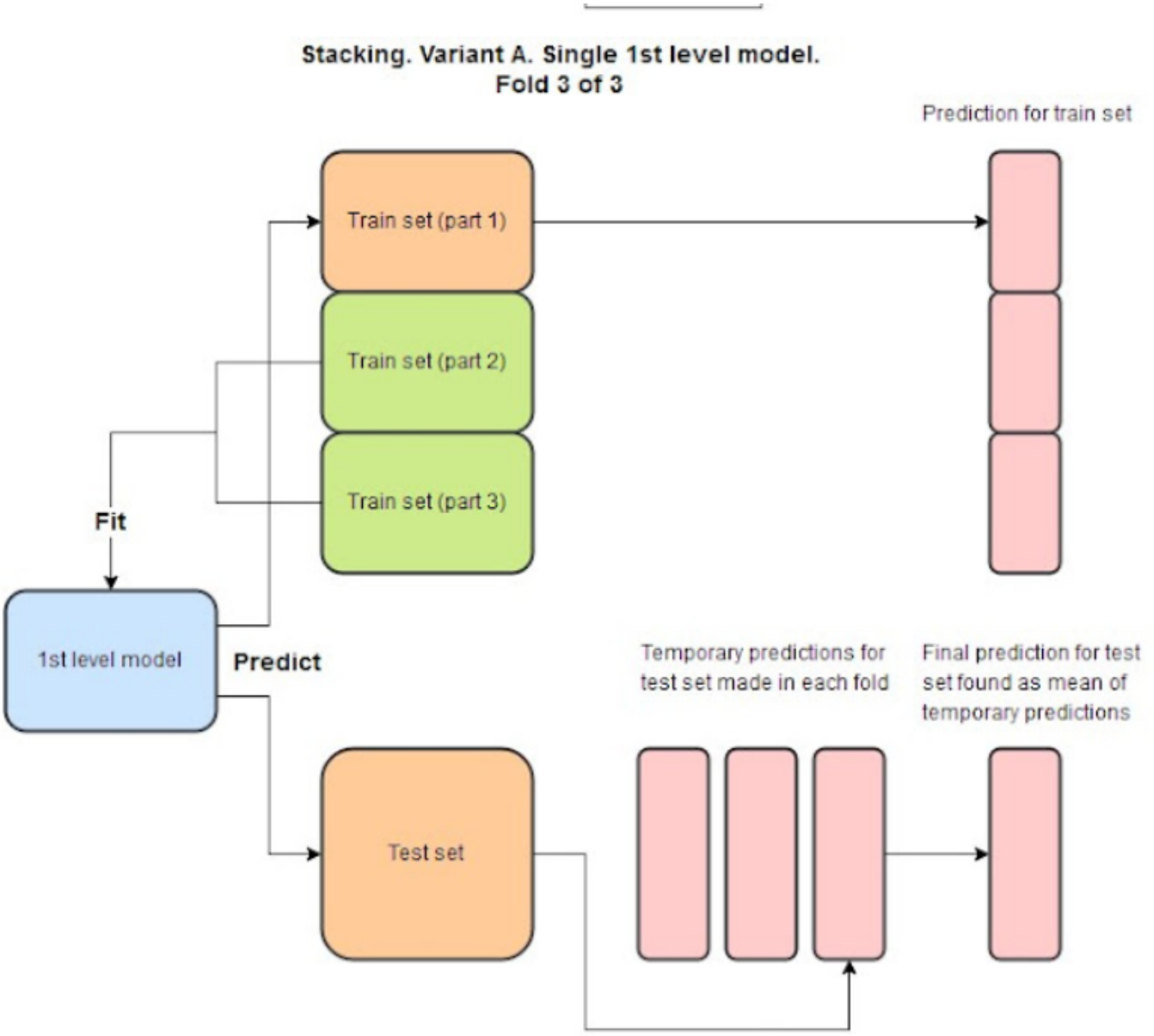
在建模的部分，我使用了一個名為**stacking**的機器學習技巧。Stacking中文稱為堆疊法，首先產生出m個 base learners(模型)彼此間並互相無關連，例如第一個 learner 為 KNN 第二個為決策樹。訓練完 m 個模型後，我們要把這m個模型合併在一起。合併的方式是我們另外再訓練一個模型，這個模型把m個base learner的輸出當成新的模型的輸入因此我們會根據這 m 個特徵利用集成式學習其中的演算

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：




Stacking. Variant A. Single 1st level model.
Fold 2 of 3
Prediction for train set
Train set (part 1)
Train set (part 2)
Train set (part 3)
Fit
1st level model
Predict
Temporary predictions for test set made in each fold
Final prediction for test set found as mean of temporary predictions
Test set




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5-Fold CV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)