Machine Learning Homework2

RE6111024 葉嘉浤

***摘要*—本次作業為實作更進階的分類器,包括 Naive Bayes Classifier, Random Forest, XGBoost, CatBoost, LightGBM,並在驗證模型表現時加入交叉驗證的步驟,檢驗模型是否足夠穩健。**

關鍵字—分類, 決策樹, Boost

I. 介紹

在分類的任務中，有許多的分類器可以使用，例如 Linear classifier, Voted perceptron, Support Vector Machine 等等，這些分類器是依據不同的參數更新方法，嘗試建立一個可以將資料分為兩類的直線或是平面。除了前述提到的較為基本的分類器，還有基於其他理論的分類器可以使用，例如 Naïve Bayes Classifier 是基於貝式定理，將事前機率轉換為事後機率，選擇事後機率最大的類別對資料分類；Random Forest 組合決策樹與投票機的概念，讓多棵決策樹進行投票，選擇票數最多的類別；建立在 Gradient boosting decision tree 之上的作法則有 XGBoost, CatBoost, LightGBM。XGBoost 的分類原理為所有決策樹的線性組合，當資料在某一棵樹較難判別時，下一棵樹會針對這些較難分的資料進行訓練；CatBoost 則是針對針對類別型特徵能有效處理，且可以使用 GPU 加速訓練；LightGBM 則是將資料結合以減少筆數，在訓練時效率非常高。本次實驗為比較這些較進階的分類器，程式碼提供在 https://github.com/chyeh1126/MachineLearning-2022。

II. 使用方法

A. *Naïve Bayes Classifier*

貝式分類器是基於貝式定理而發展出的分類器。根據貝式定理的數學式：

$$P(Y|X) = \frac{P(Y)P(X|Y)}{P(X)} \quad (1)$$

其中 X 代表一筆資料的特徵，Y 代表資料所屬的類別，貝式定理將某一個類別出現的機率(事前機率)，結合在此類別會觀察到的可能特徵值的條件機率(可能性)，轉換為觀察到某個類別時，該資料屬於某一個類別的機率(事後機率)。

Naïve Bayes Classifier 假設每個特徵彼此之間都獨立，在給定固定的特徵值，事後機率會和「事前機率與可能性」的乘積成正比，因此我們只需要計算每個類別的事前機率，乘上特徵值該類別時出現的可能性，再取最大的乘積值，就完成分類的動作。

在計算可能性的部分，如果特徵類別型或是計數型，可以直接計算給定某個類別的條件下，屬於特定值的可能性；若特徵為連續型，則需要假定在給定某個類別的條件下，該特徵會呈現的「分布」，例如常態分布、二項分布等等。

由於 Naïve Bayes Classifier 的計算方式很簡單，由於假設所有特徵彼此獨立，可以變為每一個特徵的條件機率彼此相乘，而且參數的數量不多，因此在訓練與驗證的過程非常快速，並且有數學理論的支持，可以解釋分類的結果。缺點我們需要知道每個類別的事前機率，通常會假設事前機率服從某一種分布，在假設錯誤時就會使分類效果降低。另外，當特徵之間有相關性時，並不滿足貝式定理，分類效果不好。

B. *Random Forest Classifier*

Random Forest 是一種結合 Bagging 與決策樹的算法。從原始資料依據取樣放回原則，隨機抽出固定的樣本數，根據這些樣本建立決策樹，重複此過程直到建立多個彼此獨立的決策樹。與決策樹的不同是，Random Forest 還會隨機抽出一定數量的特徵以建立決策樹，驗證時，依據每個決策樹分類的結果進行投票，票數最多的類別即為該筆資料所屬的類別。

由於 Random Forest 在樣本與特徵時都有隨機性，並使用投票的概念分類，在單一棵決策樹的分類效果不好時，可以由其他決策樹去修正，因此準確率相當高。此外，不論是連續型或是類別型特徵，Random Forest 皆能處理。缺點是需要建立很大量的樹，需要的儲存空間會比較大。

C. *XGBoost*

XGBoost 以 Gradient Boost 為基礎，在建構決策樹時，會隨機使用少量的特徵，使用 Level-wise tree growth：同一層的所有節點都生長完，才會進入下一層。而每一棵決策樹彼此具有相關性，這一棵樹無法分類的資料，下一棵樹會針對這些資料在做進一步的分類，最後建立一個較強的決策樹。此外，會在 loss function 中加入懲罰項，控制模型的複雜度。由於提供了平行計算，訓練的速度會加快，且分類問題或迴歸問題皆可使用，缺點是對空間的需求更大，因為需要儲存切割時的特徵值與每次切割時的 gradient。

D. *CatBoost*

CatBoost 是基於 Boost 的概念，能更好處理類別型特徵的算法。優點是可以將類別型特徵經過特殊處理轉換為連續型變數，同時增加維度；模型對參數值的設定不敏感，穩定性高，且支援 GPU 訓練，表示在訓練的過程可以加速。缺點為處理類別型特徵需花費較多時間與儲存空間，且容易受到隨機性的影響。

E. *LightGBM*

LightGBM 將資料以直方圖的概念作整合，將特徵作離散化，以 bin 表示資料，可以有效減少特徵數量，同時使用 GOSS 的想法：當 tree 的某一邊的 gradient 已經最佳化完成時，會轉為開始最佳化另外一邊的 gradient。LightGBM 在構築決策樹時，使用了 Leaf-wise 的概念，即持續增長 tree 的深度，如此可以降低訓練的誤差，LightGBM 可以將傳統的 Gradient boosting decision tree 的訓練速度增加數倍，且同時能有非常接近的表現。缺點是 LightGBM 所構築的決策樹會是非常傾斜的結構，並

且因為太深，可能會有過擬合的狀況，需要控制樹的深度

III. 實驗結果

本次實驗使用的資料共有 58592 筆、43 個特徵，其中有 4 個連續型特徵、28 個類別型特徵與 11 個計數型特徵，需要預測的目標為二元分類。

在資料前處理的部分，將連續型特徵依據四分位數，編碼為 0, 1, 2, 3，類別型特徵使用 Label Encoding 轉換為數值，計數型特徵則不做任何處理。

實驗設定為 80%的訓練集與 20%的測試集，比較的模型為 Naïve Bayes Classifier, Random Forest Classifier, XGBoost, CatBoost, LightGBM。評估模型的指標使用 Accuracy score 與 Recall score。在交叉驗證的部分，設定 K 分別為 3, 5, 10，切分出訓練集與驗證集，最後再對測試集做預測並評估結果。

A. 實驗數據

第一部分的實驗為讓所有模型訓練一次並比較分類的效果。

TABLE1　　各模型的預測結果

Model	Accuracy	Recall	F1-macro
Navie Bayes Classifier	0.936	0	0.485
Random Forest Classifier	0.936	0	0.485
Random Forest Classifier (sklearn)	0.932	0.008	0.489
XGBoost	0.936	0	0.483
Catboost	0.936	0	0.484
LightGBM	0.936	0	0.484

上方的表格為對所有模型訓練一次後，對測試集作預測的結果。根據表格，所有模型的 Accuracy 都很接近，都略高於 0.93，但在 Recall 的部分，則只有使用 sklearn 的 Random Forest 大於 0，其他都等於 0；F1-macro score 的部分，分數最高的也是 sklearn 的 Random Forest Classifier。雖然每一個模型的 Accuracy 很高，但 Recall 幾乎為 0，表示資料有類別不平衡的問題，模型會傾向猜測資料都是屬於數量較多的那個類別，而無法成功猜測數量較少的那個類別。

B. 交叉驗證

第二部分為在實驗加入 K-Fold 交叉驗證的機制，設定不同的 K 值，對驗證集作預測，再對多次結果取平均，作為在驗證集上的表現，最後再拿這些參數組合對測試集作預測。(手刻的 Random Forest Classifier 在此步驟錯誤，故紀錄另外 5 個模型)

a) K=3

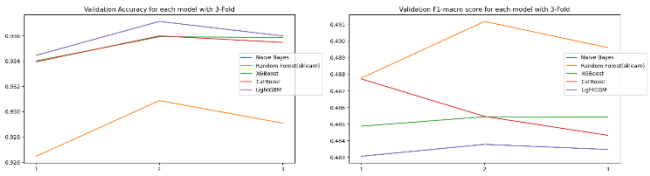

Fig.1　　各模型在 3-Fold Validation 之預測結果

上圖為 K=3 時，記錄每一次 validation 的 Accuracy 與 F1-macro score 的折線圖。根據左圖，每一個模型的 Accuracy 都在 0.93 以上，除了 sklearn 的 Random Forest Classifier 以外，另外 4 個模型的表現都很接近；根據右圖，sklearn 的 Random Forest Classifier 得到最高的 F1-macro score，表示此模型較能找到屬於類別數量較少那類的樣本，CatBoost 與 XGBoost 也可以找到少量屬於較少類別的資料，Naïve Bayes Classifier 與 LightGBM 則無法找到。

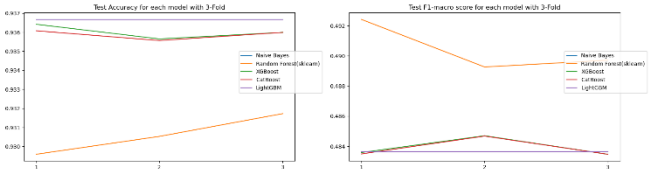

Fig.2　　各模型在 3-Fold Test 之預測結果

上圖為 K=3 時，記錄每一次 test 的 Accuracy 與 F1-macro score 的折線圖。根據左圖，我們得到的結論與在 validation 時相同；根據右圖，僅有 sklearn 的 Random Forest Classifier 能找到屬於類別數量較少那類的樣本，其他模型則相對乏力。

b) K=5

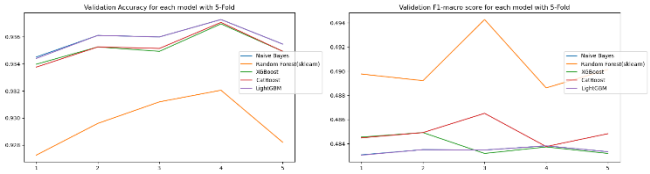

Fig.3　　各模型在 5-Fold Validation 之預測結果

上圖為 K=5 時，記錄每一次 validation 的 Accuracy 與 F1-macro score 的折線圖。根據左圖，我們可以得到與 K=3 相同的結論，Accuracy 最高的模型為 Naïve Bayes Classifer 與 LightGBM，而 sklearn 的 Random Forest Classifier 則相對低一些；根據右圖，sklearn 的 Random Forest Classifier 仍然可以找到屬於數量較少那類的資料，CatBoost 與 XGBoost 在其中幾個 Fold 也可以找到，而其他模型則無法找到。

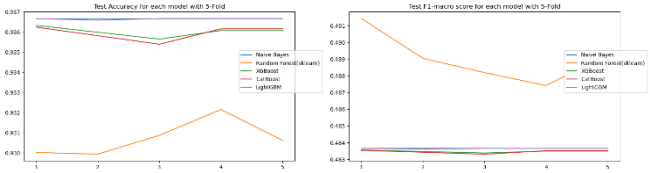

Fig.4　　各模型在 5-Fold Test 之預測結果

上圖為 K=5 時，記錄每一次 test 的 Accuracy 與 F1-macro score 的折線圖。根據圖形，Random Forest Classifier 的 Accuracy 雖然是最低的，但能找到數量較少那類的資料，其他模型則不行。

.

c) K=10

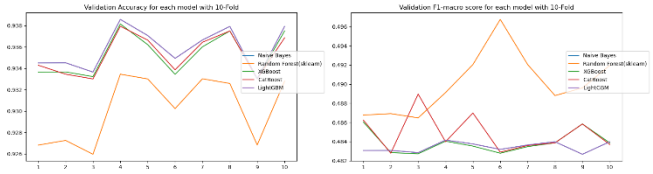

Fig.5　　各模型在 10-Fold Validation 之預測結果

上圖為 K=10 時，記錄每一次 validation 的 Accuracy 與 F1-macro score 的折線圖。在兩張圖形中，折線的波動非常劇烈，代表模型在某些 Fold 可能不穩定。根據左圖，Accuracy 最高的模型仍然是 Naïve Bayes Classifer 與 LightGBM，相對低的也仍然是 sklearn 的 Random Forest Classifier；根據右圖，也是只有 sklearn 的 Random Forest、CatBoost、XGBoost 可以找到數量較少那類的資料。

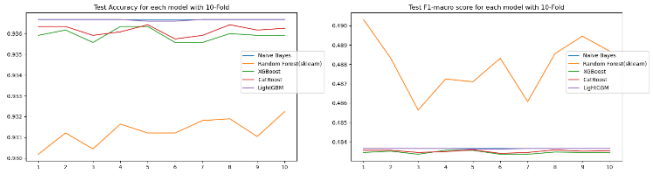

Fig.6　　各模型在 10-Fold Test 之預測結果

上圖為 K=10 時，記錄每一次 test 的 Accuracy 與 F1-macro score 的折線圖。得到的結論也與 K=3 或 5 時相同。

統整所有的實驗結果，我們發現如果資料有類別不平衡的狀況下，幾乎所有模型的 Accuracy 都很高，原因是模型會猜測數量較多的那個類別，但將評估方式換成 F1-macro score 時，得到的分數只有約 0.49，表示當資料的真實類別是較少的那類時，模型會預測錯誤。 如果是類別不平衡的資料，我們不能只參考 Accuracy，需要再參考 Recall、F1-score、Confusion Matrix 這些評估方式。

IV. 結論

在本次的實驗中，我們嘗試實作更進階的分類器，對資料作預測，並使用幾種評估方法衡量模型的表現。再第一階段的實驗，每個模型的表現都很接近。在資料有類別不平衡的情形下，每一種模型的表現都呈現一種情形：模型都會傾向猜測資料屬於數量較多的那個類別，而無法找到屬於數量較少的那個類別；在第二階段的實驗，我們加入 K-Fold 交叉驗證機制，並記錄每一組參數在 validation set 與 testing set 的分類效果，得到的結論也和第一階段相同。除了 Random Forest Classifier 之外，其他模型都嚴重受到類別不平衡的影響，可能要在切割資料時選擇分層抽樣，或是先平衡所有類別的資料量再作資料切割。