National Taiwan University **Introduction to Machine Learning and Deep Learning**
Department of Civil Engineering Instructor: C.-S. CHEN

Midterm Report
Due 23:59, Tuesday, October 31, 2023

Student ID: R12522636
Name: 吳政霖

---

1. 請說明你如何進行資料前處理。**(1%)**

1. **資料清理：**先透過 pandas 內建的 info 函式找到訓練資料或測試資料缺失項超過 90% 的欄位並加以刪除，由於非土木本科的原因，所以參考同學請教土木系朋友的結論刪除與預測補強無關的 feature，總計共刪除 7 個 feature。
2. **資料處理：**將年代異常(超過 4 位數)的欄位透過 unix 進行轉換，並將所有資料大於常態分布 90%數值的 20 倍的值設為 NaN，負數則改為 0。接著進行缺項值填補，採用的作法為 IterativeImputer 而 estimator 使用的是 BayesianRidge，補完缺失值之後進行 Scaler 處理，採用的做法為 RobustScaler，並用預設的 25%~75%區間參數。
3. **進行特徵篩選：**使用 XGBClassifier 模型並採用 permutation_importance 的方法最後挑選出 8 項重要特徵。
4. **重新處理原始資料：**針對重要的 8 筆特徵重新 reload 原始資料，而重要特徵已捨去年代資料故不進行處理，接著將特徵重複進行第三步驟的 Imputer 及 Scaler 處理。
5. **Oversampling 處理：**在資料處理的最後使用 SMOTE 方法來改善 data imbalance。

2. 請說明你所使用的 **model** 以及 **hyper-parameter tuning** 的心得。**(1%)**

在特徵篩選階段採用的模型是 XGBClassifier，由於網路上關於 permutation_importance 的資料說明這個挑選 feature 的方法較不影響最終的預測模型，而是更重視資料之間的關係，所以這邊就使用簡單的 XGBClassifier 來進行挑選。
而最終的預測模型採用的是 XGBClassifier、ExtraTreesClassifier 及 RandomForestClassifier 三者作為 VotingClassifier 的模型參數。在調參的過程真的是相當坎坷，由於一開始不知道要用什麼方式來找參數，所以就一股腦的都用 Gridsearch 的方法，有時候跑半天的結果卻不如原始模型只設定 randomstate=42 來的好，後來逐漸看到有些人的做法是會先用 Randomsearch 的方式隨機的大範圍搜索相對好的參數，再針對這些參數以區域性的範圍擴張做 Gridsearch，這個方法可以有效的節省運算資源及時間，也可以找到相對較佳的解法，但我自己不管是採用哪個方法做出來的效果都不如預期，終於能體會到網路上人家說 ML 就是一門通靈學的心情。

3. 畫出 **confusion matrix** 分析 **model** 分類的結果，並列出 **precision**、**recall** 和 **F1-score**，再加以簡單說明。**(2%)**

以下混淆矩陣表格以訓練資料拆分 20%作為測試資料產出

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	TP=475	FN=39
	Class=No	FP=71	TN=481

Results：

- Class balance: Yes : No = (475+39) : (71+481) = 514:552
- Accuracy: ACC (TP+TN)/(TP+TN+FP+FN) = (475+481)/(475+481+71+39) = 0.8968
- Precision: P = TP/(TP+FP) = 475/(475+71) = 0.8699
- Recall: R = TP/(TP+FN) = 475/(475+39) = 0.9241
- F-measure: F = 2RP/(R+P) = 2*0.9241*0.8699/ (0.9241+0.8699) = 0.8962
- False-alarm rate: FNR = FN/(FN+TP) = 39/(39+475) = 0.0758
- Misdetection rate: FPR = FP/(FP+TN) = 71/(71+481) = 0.1286

Description：

在Class balance中可以看到對於1跟0的預測是趨近於平衡的，雖然在不平衡的訓練資料及測試資料中預期會產出不平衡的類別，但我的猜測是由於在資料處理中有使用SMOTE方法來增加模型預測0的機率，最終改善模型預測的平衡性。

而上述的混淆矩陣分析結果中，精確率Precision代表「預測Yes中真正Yes的比例」，而查全率Recall代表「預測Yes總共佔了真正Yes多少比例」。Precision的數據約為0.87，也就是在所有被模型預測為"Yes"的樣本中，有約87%是真正屬於"Yes"的，而 Recall約為0.92，也就是模型成功捕捉到了約92%的實際"Yes"樣本。

準確率Accuracy為0.8968，表示模型正確預測的樣本數占總樣本數的比例，而F1分數為0.8962，是一個綜合性能指標，並且接近於準確率，表示模型能夠有效的區分正類別和負類別。此外，偽警報率0.0758和誤檢率0.1286也相對較低。

Submission Format

Convert `midterm_report_template.docx` to `midterm_report.pdf`, then place `midterm_report.pdf` and `codes` into a folder named {yourStudentID}_midterm and compress it into a ZIP file for upload to NTU COOL. Below is the file format example for upload.