Machine Learning – Assignment I

All about supervised learning

黃亮臻 RE6124035
Institute of Data Science, NCKU

關鍵字—Linear Classifier, K-NN, Decision Tree, feature importance, SHAP, Cross-Validation.

I. 介紹

本次作業分為三大部分，分別為：1. 分類模型 2. 特徵工程 3. 交叉驗證。第一部分需手刻實現 Linear Classifier, K-NN Classifier, Naïve Decision Tree Classifier 以及 Decision Tree with Pruning; 第二部分需使用 Linear Classifiers, Decision Tree 以及 SHAP 計算特徵重要性，並使用挑選出的重要變數嘗試提高模型表現; 第三部分使用交叉驗證，檢驗模型的穩健性與是否過度擬合。

II. 使用方法

A. Linear Classifier

訓練一組權重 $W = (w_1, w_2, \dots, w_n)$，對資料做線性組合相乘後，若$WX >= 0$則預測為 1 (True), $WX < 0$則預測為 0 (False)。模型的訓練為逐個樣本進行，當預測錯誤時，會通過調整權重和誤差來更新模型。另外會設置 lr (學習率) 控制模型學習時參數更新的變化量，以及 epoch (迭代次數) 控制模型在訓練集上的迭代次數。另外，最後訓練得到的權重可作為解釋變數的依據。

B. K-NN Classifier

K-近鄰算法（K-Nearest Neighbors，簡稱 K-NN），原理為藉由找出新資料的最近 K 個鄰居，再根據這些鄰居的類別，以多數決的方式來預測新資料的類別。而計算鄰居距離的方式有很多種，在本次實驗中，可選擇使用歐式距離、曼哈頓距離、餘弦相似度來尋找鄰居。

C. Naïve Decision Tree Classifier

決策樹的核心觀念為不斷設置條件，將原始的資料集分割成越來越小的子集，直到最後達到的決策目的。首先，需要找出好的分割節點，好的節點的概念為分割後的子集比分割前更乾淨，即分割後的組內樣本更相似。常見作為評估分割好壞的指標有：Information Gain、Gini Index，兩者皆有計算純度的概念，值越小表示亂度小，越集中在某個類別上。因此若分割後的節點其 Information Gain/Gini Index 較分割前小，表示這是一個可用的節點，而使得 Information Gain/Gini Index 減少最多的節點則為最佳分割點，減少的量也會作為計算 feature importance 的依據。

另外，選擇最佳分割點時，較正規的做法是每一次都將所有變數的所有唯一值當作分割點計算 Information Gain/Gini Index，再選出 Information Gain/Gini Index 最小的變數與對應的值作為這一次的分割點。但由於計算量過於龐大，我的寫法是若種類數量大於十種，則只取前十種計算 Information Gain/Gini Index，可以稍微避免掉連續變數將每個唯一值都計算一次，運算量龐大也沒有必要的情形。

D. Decision Tree with Pruning

為了避免過度擬合與加快訓練速度，決策樹可能需要進行剪枝，即將基礎的決策樹模型設置停止條件。在本次實驗中，分別為設了: max_depth (最大深度) 與 min_samples_split (分割內部節點所需的最小樣本數) 兩種剪枝方式，當樹達到設定的深度或當節點的樣本數小於設定值時，將停止分割。

E. SHAP

SHAP (SHapley Additive exPlanations) 為一種與原模型無關的解釋方法，通過計算每個特徵對預測的貢獻，來解釋個體實例的預測，衡量特徵對最終預測值的影響。其公式為：

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j \tag{4.2}$$

g 為解釋模型；$z' \in \{0,1\}^M$ 表示特徵是否存在；M 為特徵的個數；ϕ_0 為模型預測平均值；ϕ_j 為特徵 j 的 Shapley value。

Shapley value 為欲計算之特徵在所有可能的特徵集合中的平均邊際貢獻。其公式如下：

$$\phi_j(val) = \sum_{S \subseteq \{x_1, x_2, \dots x_p\}/\{x_j\}} \frac{|S|!\,(p-|S|-1)!}{p!} \Big(val(S \cup x_j) - val(S) \Big) \tag{4.3}$$

S 為模型中使用的特徵子集；$\boldsymbol{x}$ 為要解釋的實例的特徵值向量；p 為特徵數量。

若對單個實例做 SHAP，可由每個特徵的 Shapley value 得知各個特徵對於此實例的重要性。若將每個實例都進行 SHAP，即可獲得每個實例與每個特徵的 Shapley value 矩陣，可通過分析此矩陣試著解釋整個模型。

III. 實驗結果

本次實驗使用的資料有 58592 筆、43 個變數。其中包含 28 個類別變數、15 個數值型變數，預測目標為二元分類。

資料前處理的部分，我將類別變數做 label encoding 轉為數值型態，並將資料切分成 80%訓練集與 20%測試集。比較的模型為：Linear Classifier, KNN Classifier, Naïve Decision Tree Classifier, Decision Tree with Pruning，評估模型的指標我選擇使用 Accuracy 與 F1-score，並在最後做 3 Fold, 5 Fold, 10 Fold 交叉驗證，評估模型的表現。

A. 實驗數據

第一部分為讓所有模型訓練一次，測試集在模型上的表現。

Table 1 各模型的預測結果

Model	Accuracy	F1-score	Time
Linear Classifier	0.936	0.484	57.5 s
KNN Classifier	0.934	0.485	69 s
Naïve Decision Tree Classifier	0.870	0.508	22.6 s
Decision Tree with Pruning	0.933	0.488	4.35 s

由上方 Table1 可知，就 Accuracy 而言，線性分類器、KNN、剪枝過的決策樹表現都很好，未剪枝 Decision Tree 也有一定的水準；但從 F1-score 來看，每個分類器的表現都不好。那是因為資料有不平衡的問題，模型傾向猜測數量較多的類別，經檢查也發現正樣本僅佔資料的 6.4%。特別的地方是，若是從 Accuracy 觀察，Naïve Decision Tree Classifier 在各模型中的表現稍差，但 F1-score 卻是各個模型的最高分，推測 Decision Tree 的分類方式相較其他模型更可以容許不平衡的資料。

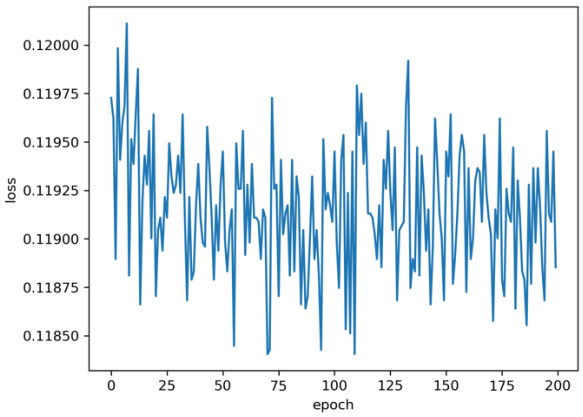

Figure 1 Linear Classifier Epoch-Loss 關係圖

由上方 Figure 1 可知，Linear Classifier 的 Loss 並沒有隨著迭代次數越多而降低，推測模型沒有學到東西。若希望訓練出好的線性分類器，可能需要先處理資料不平衡的問題。

B. 特徵工程

1) Feature importance

上述模型中，Linear Classifier 訓練出的權重，以及 Decision Tree 選擇最佳分割點時所計算的 Feature Importance，可以一定程度反映變數的重要程度。

由於 Decision Tree 模型內計算 Feature Importance 的方式為將其縮到[0, 1]之間，為了方便比較，我將 Linear Classifier 訓練出的權重取絕對值，也將其縮到 [0, 1] 之間。下圖為 Linear Classifier, 剪枝與未剪枝的 Decision Tree 對於各個變數的重要程度：

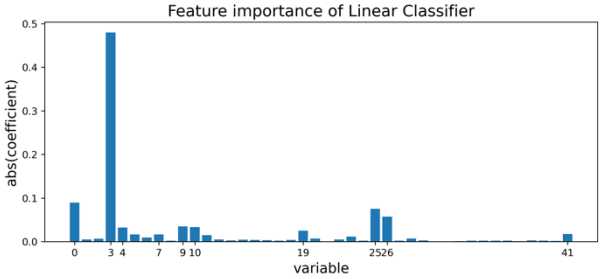

Figure 2 Feature Importance of Linear Classifier

Figure 2 為將線性分類器的係數取絕對值，並縮到 [0, 1] 之間的結果。最重要的 10 個變數為: area_cluster、policy_tenure、length、width、max_torque、max_power、population_density、displacement、ncap_rating、model。

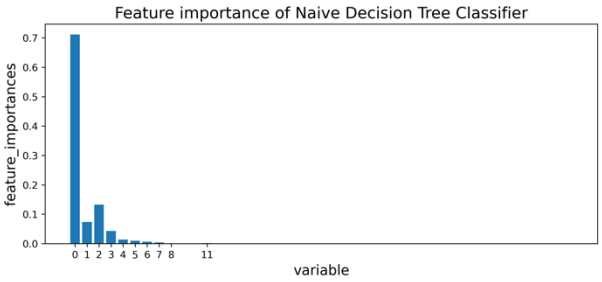

Figure 3 Feature Importance of Naïve Decision Tree Classifier

Figure 3 為 Naïve Decision Tree Classifier 的 Feature Importance。最重要的 10 個變數為: policy_tenure、age_of_policyholder、age_of_car、area_cluster、population_density、make、segment、model、engine_type、fuel_type。

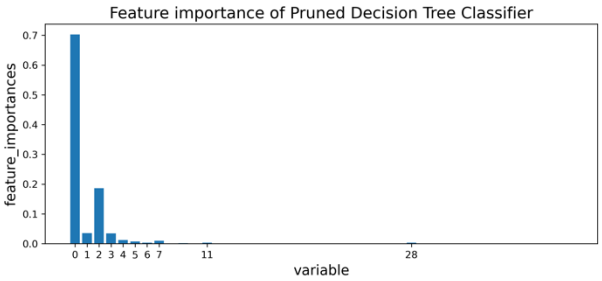

Figure 4 Feature Importance of Decision Tree with Pruning

Figure 4 為 Decision Tree with Pruning 的 Feature Importance。最重要的 10 個變數為: policy_tenure、age_of_policyholder、age_of_car、area_cluster、population_density、model、make、segment、engine_type、gross_weight。

2) SHAP

SHAP 為一種解釋機器學習預測的方法，下圖為 shap values 最大的十個變數：

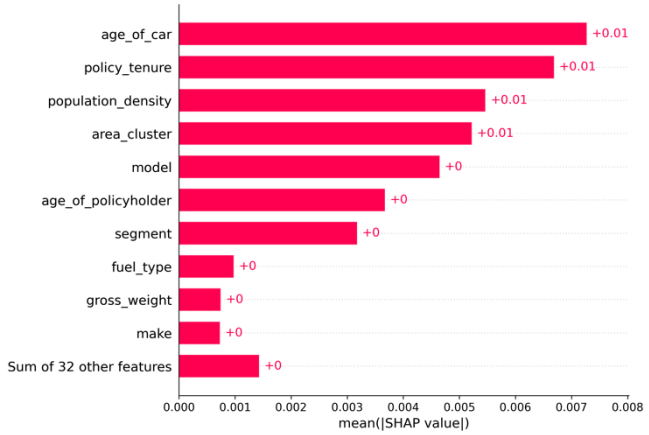

Figure 5 SHAP values of Decision Tree with Pruning

Figure 5 為使用 Decision Tree with Pruning 的預測拿去做 SHAP 的結果。shap value 最大的十個變數為：age_of_car 、 policy_tenure 、 population_density 、 area_cluster 、 model 、 age_of_policyholder 、 segment 、 fuel_type 、 gross_weight 、 make。其中與 Pruned Decision Tree 根據 Feature Importance 挑出的變數幾乎相同，只有 engine_type 沒有進入 SHAP values 的前十名。

特別的一點是，SHAP values 的長條圖看起來較 Feature Importance 的長條圖平緩一些，推測 Feature Importance 的計算方式容易讓少數重要變數獨佔，遇到某些情況可能會較難抓出貢獻小一點，但其實也有一定解釋能力的變數。

3) Designing new features

我將以上模型透過 Feature Importance 選出的十個變數，與 SHAP values 最大的十個變數取聯集，共留下 17 個變數，再使用這些變數訓練一次模型，結果如下：

Table 2 挑選重要特徵後的模型預測結果

Model	Accuracy	F1-score	Time
Linear Classifier	0.936	0.484	47.9 s
KNN Classifier	0.934	0.487	61.5 s
Naïve Decision Tree Classifier	0.871	0.504	19.6 s
Decision Tree with Pruning	0.932	0.487	4.1 s

將 Table 2 與 Table 1 做比較會發現，四種模型的表現差不多，雖然沒能提高正確率，但訓練時間縮短了不少，因此後續的交叉驗證我會使用保留重要特徵的新資料訓練模型。

C. 交叉驗證

在實驗中加入 K-Fold 交叉驗證，設定不同的 K 值（本次實驗設定 K=3, 5, 10）對驗證集做預測，再對每個 Fold 的預測結果取平均，作為驗證集上的表現，最後再拿訓練過的參數組合對測試集做預測。

1) K = 3

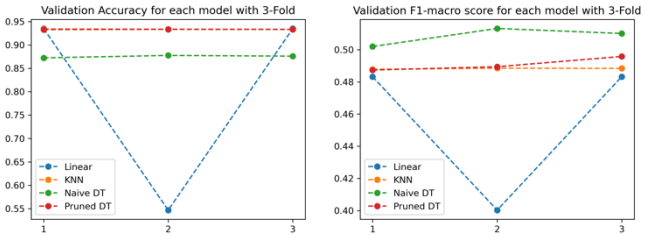

Figure 6 各模型在 3 Fold Validation 的預測結果

上圖為 K = 3 時，紀錄每一次 validation set 的 accuracy 與 F1 score 的折線圖。由圖來看，線性分類器的結果非常不穩定，第二個 Fold 的結果突然掉非常多。就 accuracy 而言，KNN 與剪枝過的 Decision Tree 表現很好，但從 F1-score 來看，沒有剪枝的 Decision Tree 表現最好，與 table 1 推測的結果相同。

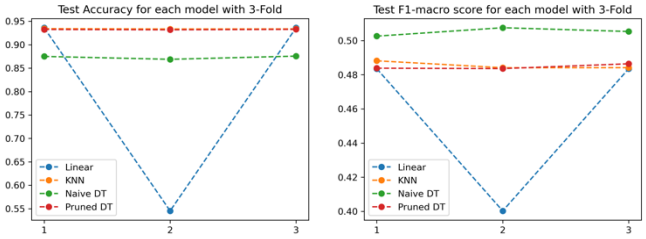

Figure 7 各模型在 3 Fold Test 的預測結果

上圖為 K = 3 時，紀錄每一次 testing set 的 accuracy 與 F1 score 的折線圖。其表現與 validation set 差不多，表示模型的泛化能力還不錯，也沒有過度擬合的現象。

2) K = 5

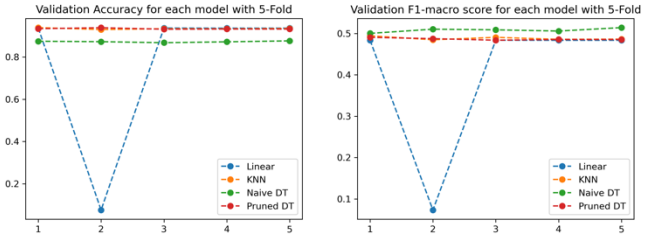

Figure 8 各模型在 5 Fold Validation 的預測結果

K = 5 時各個模型的表現與 K = 3 幾乎相同，同樣是線性分類器較不穩定，從 Accuracy 來看，剪枝的 Decision Tree 與 KNN 的表現最好，從 F1-score 來看，沒有剪枝的 Decision Tree 較好。

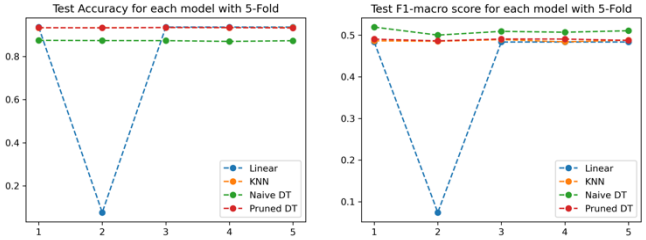

Figure 9 各模型在 5 Fold Test 的預測結果

上圖為 K = 5 時，紀錄每一次 testing set 的 accuracy 與 F1 score 的折線圖。其表現與 validation set 幾乎相同，表示模型的泛化能力還不錯，也沒有過度擬合的現象。

3) K = 10

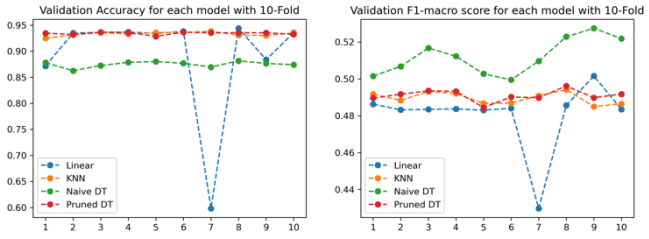

Figure 10 各模型在 10 Fold Validation 的預測結果

K = 10 時各個模型的表現與 K = 3, 5 時幾乎相同，同樣是線性分類器較不穩定，從 Accuracy 來看，剪枝的 Decision Tree 與 KNN 的表現最好，從 F1-score 來看，沒有剪枝的 Decision Tree 較好。

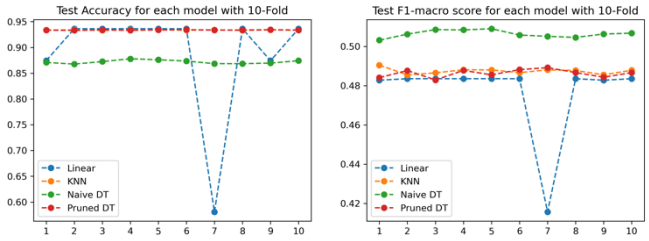

Figure 11 各模型在 10 Fold Test 的預測結果

上圖為 K = 10 時，紀錄每一次 testing set 的 accuracy 與 F1 score 的折線圖。其表現與 validation set 相差不大，表示模型的泛化能力還不錯，也沒有過度擬合的現象。

IV. 結論

在本次作業中，使用了 Linear Classifier、K-NN Classifier、Naïve Decision Tree Classifier 以及 Decision Tree with Pruning 對資料做預測。在第一階段的實驗中，Linear Classifier、K-NN Classifier、Decision Tree with Pruning 的 Accuracy 都很接近，Naïve Decision Tree Classifier 則稍微低了一些，但 Naïve Decision Tree Classifier 的 F1-score 卻相較其他模型高；第二階段的實驗中，僅使用 SHAP 與第一階段模型挑選出的重要變數做預測，模型表現與第一階段差不多，訓練速度也縮短了不少；第三階段的交叉驗證中，發現了 Linear Classifier 的不穩定性，其餘模型的表現則與前兩個階段相差不大。

模型的 Accuracy 很高，有很大的原因跟資料類別不平衡有關，由於測試集的資料也有類別不平衡的情形，因此當模型傾向預測數量多的類別時，正確率看似會跟著提高，但並不代表模型有學到東西。如果是類別不平衡的二元分類資料，不能只看 accuracy，同時參考 F1-score 指標會比較妥當。