FAI HW3 Report

姓名: 鐘文駿
學號: b11902167

(a)

Linear Model

Classification:

1. 初始化 W 為 $n_{features} * n_{classes}$ 且值全部為 0 的矩陣，根據 model 的 $iterations$ 次數計算每次的 descent gradient，並逐次更新 W 內的值。
2. 每次更新 W 時，計算當下每個物件預測是哪個 classes 的機率 $Y_{pred} = X_{train} * W$ (使用 `_softmax()` 使機率總合為 1)，使用 one-hot 的矩陣 Y_{true} 去計算兩個矩陣的 differences $Y_{diff} = Y_{true} - Y_{pred}$。Gradient (normalize後) 會等於 cross-entropy loss 的 partial derivative，也就是

$$Gradient = -\frac{X_{train}^T Y_{diff}}{n_{samples}}$$

3. 依 step 1、2 可以計算出最終每個 classes 對不同 feature 的 weight: $W_{n_{features} * n_{classes}}$，完成 fit。
4. Predict 時，$Y_{pred} = X_{test} * W$ 可以計算出每個物件預測對每個 classes 的機率，機率最高的 class 即為預測的答案。

Regression:

1. 初始化 W 為 $n_{features}$ 且值全部為 0 的矩陣，根據 model 的 $iterations$ 次數計算每次的 descent gradient，並逐次更新 W 內的值。
2. 每次更新 W 時，計算 $Y_{pred} = X_{train} * W$，並計算與真實的值的 differences $Y_{diff} = Y_{true} - Y_{pred}$。計算 gradient 的方法與 Classification 相同。
3. 依 step 1、2 可以計算出最終每個 classes 對不同 feature 的 weight: $W_{n_{features}}$，完成 fit。
4. Predict 時，$Y_{pred} = X_{test} * W$ 可以計算出每個物件最終預測的值。

Decision Tree Model

Classification:

1. 對一群 samples (初始為 X_{train})，用 `_find_best_split()` 去尋找最適合的 feature 和相對應的 threshold。

2. 在 `_find_best_split()` 中，目的是尋找 Gini index 最小的 feature。計算 Gini index 的方式為，根據目前所分割的左子樹和右子樹。分別去計算每個 class 出現的機率，並計算左右 Gini index 分別的值，接著根據左右子樹的 sample 數量去計算加權平均:

$$Gini = 1 - \sum_{i=1}^{k} p_i^2$$

3. 找到最好的 feature 和其 threshold 後，用它來分這群 sample 的左子樹和右子樹。
4. 重複 step 1~4，直到這棵樹的深度等於 max_depth 為止。當達到終止條件時，建立 leaf node，其值為這群 sample 中佔比最大的 class。
5. Predict 時，把每個 data 的 feature vector 丟入 `_traverse_tree()`，檢查這個 data 屬於哪一個 class。

Regression:

1. 步驟與 classification 相同，但在 `_find_best_split()` 時會分割 MSE 最小 feature 和其 threshold。
2. 建立 leaf node 的值為這群 sample value 的平均。
3. Predict 時與 classification 相同，得出其預測值。

Random Forest Model

Classification:

1. 建立 $n\_estimator$ 個深度為 max_depth 的 Decision Tree。
2. 依序對每個樹，隨機從 X_{train} 裡面取 samples (可重複)。把這些 samples 丟入當前的 tree 做 fit。
3. Predict 時，把當前要預測的 sample 丟入所有 Decision Tree 中，佔最多數的 class 即為預測結果。

Regression:

1. 前兩步驟與 classification 相同。
2. Predict 時，把當前要預測的 sample 丟入所有 Decision Tree 中，所有 Decision Tree 結果的平均即為所求。

Correctness

我根據已知 Model 的方法進行實作，並且調整參數後預測的結果有變好。且因為我們已經知道 Y_{test}，所以我們可以簡單的算出 Accuracy。並且還根據不同參數的情況下，漸漸減少誤差或提高準確性，所以可以確定實作的方法是對的。

(b)

以下是不同 Model 分別的 Accuracy 和 MSE:

- Logistic Regression Accuracy: 0.8888
- Decision Tree Classifier Accuracy: 0.8222
- Random Forest Classifier Accuracy: 0.8888
- Linear Regression MSE: 22.4103
- Decision Tree Regressor MSE: 18.0046
- Random Forest Regressor MSE: 10.9822

1. Random Forest 表現的比其他兩種 Model 好 (更穩定)，因為他用了 ensembling learning 和減少 Variance。
2. Decision Tree 會 overfit，尤其在樹的深度比較深的時候。
3. 給的 dataset 沒有 linearly seperable，所以表現的比其他兩個 Model 還差。

(c)

用 Random Forest Regression 做例子:

- MSE in normalization: 23.1227
- MSE in standardization: 10.9822

Normalization 有可能會把重要的 feature 與相對不重要的 feature 有相同的 weight，standardization 會擁有較好的 gradient descent。而且我發現在多數情況下， standardization 比 normalization 表現得更好，但還是有 normalization 比較好的情況。彼此感覺沒有什麼明顯的缺點或優點，還是要根據 dataset 去做調整。

(d)

- Logistic regression model with 1000 iterations, learning rate 0.01, accuracy is 0.8888
- Logistic regression model with 10000 iterations, learning rate 0.01, accuracy is 0.9111
- Logistic regression model with 5000 iterations, learning rate 0.1, accuracy is 1

調到適當的 iteration 數可以讓 accuracy 更高，應該是會 converge 的關係。調到適當的 learning rate 可以更快的 converge，但是太高會讓整個 model diverge，所以還是要適當的調整參數找到最好的結果。

(e)

1. 樹的數量越多，越有機會增加 model 的 accuracy，但會讓 complexity 上升。
2. 樹的深度越大，會減少 model 的 bias 但會有 overfitting 的風險
3. 我選擇用 150 個深度為 5 的 Decision tree，可以使 model 不會跑太久的同時也保持一定的 accuracy

(f)

Linear Model:

- strengths: 快速並且 variance 較低，非常容易理解
- weaknesses: 在不是 linear seperable 的 dataset 裡表現得很差

Decision Tree Model:

- strengths: 可以簡單的解決 non-linear 的 dataset
- weaknesses: 有 overfitting 的風險

Random Forest Model:

- strengths: Accuracy 通常較高、較穩定，且可以解決 overfitting 的問題
- weaknesses: 需要花費大量的時間 train model，理解整個 model 的運作較困難

我覺得 Random Forest Model 雖然 train 會需要大量的時間，但在這種沒有時間限制的情況裡他然是最適合的選擇，擁有在大多情況下較高、較穩定的 accuracy。而且他也可以解決 Decision Tree 會友的 overfitting 的問題，並且也可以處理 non-linear 的 pattern，所以會傾向使用這個較準確的 model。