團隊測驗報告

報名序號：110009

團隊名稱：熊熊一定行

註1：請用本PowerPoint 文件撰寫團隊程式說明，**請轉成PDF檔案繳交**。
註2：依據競賽須知第七條，第4項規定：
測試報告之簡報資料不得出現企業、學校系所標誌、提及企業名稱、學校系所、教授姓名及任何可供辨識參賽團隊組織或個人身分的資料或資訊，違者取消參賽資格或由評審會議決議處理方式。

一、資料前處理(說明資料前處理過程)

Step1:確認資料品質
- 資料筆數
- 遺失值
- 資料型態
- 資料的特徵數量（欄位數量）

Step2:資料轉換與合併
- 將F_1到F_13做正規化
- 正規化:x’ =(x-min(x))/(max(x)-min(x))
- 正規化後的資料跟原始資料做合併
- 原始資料的F_1裡面有出現數值為0平移到1

Step3:特徵工程-衍生出新變數
- 將正規化後的變數取指數、平方、立方
- 將原始資料的變數取根號、log、sin、cos、倒數

Step4:交互作用-衍生出新變數
- 將原始變數、正規化後的變數和特徵工程新變數做交互作用

Step5:減少資料量
- 透過Lasso變數篩選法

二、演算法和模型介紹(介紹方法細節)

• Lasso Regression 變數篩選法

- 在進行迴歸分析時，我們將要預測的變數當作反應變數，剩下的變數當作解釋變數，透過迴歸分析我們可以去預測反應變數，但這裡會有一個問題，過多的解釋變數，會造成我們SSE提升，造成Overfitting的結果，這種情況是我們非常不願意樂見的，另外一種狀況就是資料本身具有高度共線性，資料之間具有高度的相關，造成迴歸係數的不穩定，這也是我們不想樂見，綜合上述兩點高度共線性以 及Overfitting的問題，是高度影響預測結果的原因。

- 因此要避免上述兩點原因，我們可以藉由Lasso Regression來避免，Lasso Regression為一種同時進行特徵工程和正則化的迴歸分析方法，主要目的為增強統計模型的預測準確性和可解釋性。Lasso Reg -ression是由Regularized Regression所演化爾來，其目標函數與OLS Regression相同，但多了一個稱 為 Penalty parameter的參數為$minimize\{SSE + P\}$，而這個參數分別對應兩種分別為L1 Penalty以及L2 Penalty，而Lasso正好對應為L1 Penalty $minimize\{SSE + \lambda \sigma_{j=1}^{p} |\beta_j|\}$，Lasso Regression使用Regulariztion來優化模型，同時也具有變數篩選的功能，因此可以避免上述兩種原因的產生，故我們在這裡利用之。

二、演算法和模型介紹(介紹方法細節)(續)

XGBoost演算法用來建立模型

XGBoost (Extreme Gradient Boosting)算法思想:

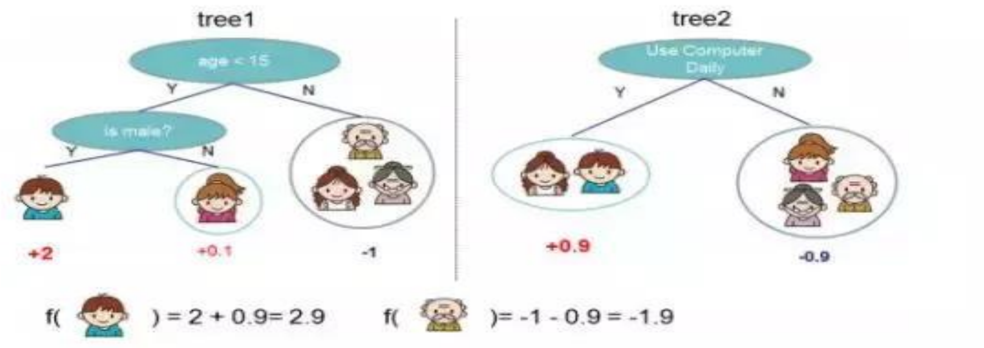
該算法思想就是不斷地添加樹，不斷地進行特徵分裂來生長一棵樹，每次添加一個樹，其實是學習一個新函數，去擬合上次預測的殘差。當我們訓練完成得到k棵樹，我們要預測一個樣本的分數，其實就是根據這個樣本的特徵，在每棵樹中會落到對應的一個葉子節點，每個葉子節點就對應一個分數，最後只需要將每棵樹對應的分數加起來就是該樣本的預測值。

$$\hat{y} = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i)$$

$$where\ F = \{f(x) = w_{q(x)}\}(q : R^m \rightarrow T, w \in R^T)$$

註：w_q(x)為葉子節點q的分數，f(x)為其中一棵回歸樹

如下圖例子，訓練出了2棵決策樹，小孩的預測分數就是兩棵樹中小孩所落到的結點的分數相加。爺爺的預測分數同理。



二、演算法和模型介紹(介紹方法細節)(續)

XGBoost原理

XGBoost目標函數定義為：

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \qquad where\ \Omega(f) = \gamma T + \frac{1}{2}\lambda||w||^2$$

Training loss　　Complexity of the Trees

目標函數由兩部分構成，第一部分用來衡量預測分數和真實分數的差距，另一部分則是正則化項。正則化項同樣包含兩部分，T表示葉子結點的個數，w表示葉子節點的分數。γ可以控制葉子結點的個數，λ可以控制葉子節點的分數不會過大，防止過擬合。

正如上文所說，新生成的樹是要擬合上次預測的殘差的，即當生成t棵樹後，預測分數可以寫成：

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i)$$

同時，可以將目標函數改寫成：

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t)$$

二、演算法和模型介紹(介紹方法細節)(續)

很明顯，我們接下來就是要去找到一個f_t能夠最小化目標函數。XGBoost的想法是利用其在f_t=0處的泰勒二階展開近似它。所以，目標函數近似為：

$$\mathcal{L}^{(t)} \simeq \sum_{i=1}^{n}[l(y_i, \hat{y}^{(t-1)}) + g_i f_t(\mathbf{x}_i) + \frac{1}{2}h_i f_t^2(\mathbf{x}_i)] + \Omega(f_t)$$

其中g_i為一階導數，h_i為二階導數：

$$g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)}), \quad h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$$

由於前t-1棵樹的預測分數與y的殘差對目標函數優化不影響，可以直接去掉。簡化目標函數為：

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n}[g_i f_t(\mathbf{x}_i) + \frac{1}{2}h_i f_t^2(\mathbf{x}_i)] + \Omega(f_t)$$

二、演算法和模型介紹(介紹方法細節)(續)

上式是將每個樣本的損失函數值加起來，我們知道，每個樣本都最終會落到一個葉子結點中，所以我們可以將所以同一個葉子結點的樣本重組起來，過程如下圖：

$$
\begin{aligned}
Obj^{(t)} & \simeq \sum_{i=1}^{n}\left[g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)\right] + \Omega(f_t) \\
& = \sum_{i=1}^{n}\left[g_i w_{q(x_i)} + \frac{1}{2}h_i w_{q(x_i)}^2\right] + \gamma T + \lambda\frac{1}{2}\sum_{j=1}^{T} w_j^2 \\
& = \sum_{j=1}^{T}\left[(\sum_{i\in I_j} g_i)w_j + \frac{1}{2}(\sum_{i\in I_j} h_i + \lambda)w_j^2\right] + \gamma T
\end{aligned}
$$

因此通過上式的改寫，我們可以將目標函數改寫成關於葉子結點分數w的一個一元二次函數，求解最優的w和目標函數值就變得很簡單了，直接使用頂點公式即可。因此，最優的w和目標函數公式為:

$$
w_j^* = -\frac{G_j}{H_j+\lambda} \quad Obj = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda} + \gamma T
$$

三、預測結果

利用Lasso篩選變數出來的變數再用Xgboost建立模型。

編號	預測值
1	11.90205574
2	11.90205574
3	11.90205574
4	11.90205574
5	11.90205574
6	11.90205574
7	11.90524101
8	11.90205574
9	11.90524101
10	11.90524101
11	11.90524101
12	11.90205574
13	11.90205574
14	11.90205574
15	11.90205574
16	11.90205574
17	11.90205574
18	11.90205574
19	11.90205574
20	11.90205574