題目：台灣牧場乳量預測

組員：洪廷維、林宏宇、趙柏鈞

1. Introduction

本專題的目的是實現牧場乳量預測，我們被給予四份關於乳牛的資料庫：birth.csv、breed.csv、report.csv、spec.csv，接著必須自己進行資料處理，篩選出有用的資料，並利用機器學習的方式預測台灣不同地區牧場生產的乳量，以掌握乳量生產的關鍵。

1. Framework
   1. 概述

本專題使用Python及其相關套件（numpy, pandas, keras, xgboost, scikit-learn）實現乳量預測。軟體主架構如圖1所示，可將整體分成兩個部分：資料前處理、機器學習模型，首先會透過資料前處理將四份資料庫轉為Model Inputs（training data及test data），接著用training data訓練模型，最後把test data丟進訓練好的模型以產出預測結果。

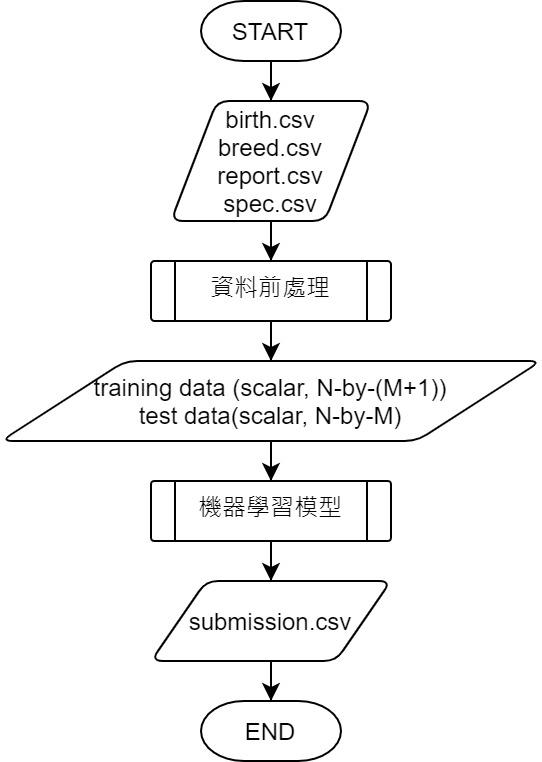


圖1

* 1. 資料前處理

資料前處理的目的是提取資料庫中有用的資料，使預測準確度提升，其基本概念可分為資料清理、資料整合、資料轉換，下文會分別介紹。資料前處理架構如圖2所示，一開始先讀檔，接著每次選一個特徵進行資料整合，直到沒有有用的特徵後進行資料轉換，最後再拆成training data、test data。

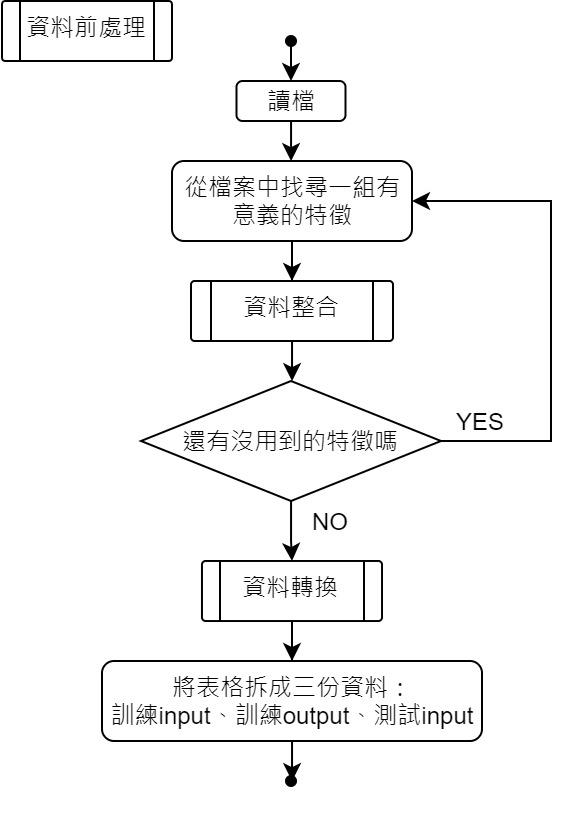


圖2

* + 1. 資料清理

資料清理是指清除與乳量無關聯的特徵，或是清除、填補缺漏的資料，由於資料清理是很基本的動作，在每個步驟都可能執行，所以未在圖2中畫出。我們將資料清理分為橫向（一個特徵）與縱向（一筆資料）清理。

橫向清理是對特徵作清理，首先我們利用產業知識判斷該特徵對乳量多寡的重要性，這部分參考了農委會[1]、維基百科、各式部落格[2]網站等等，記載了大量關於影響乳牛產乳量的知識，其次也參考相關乳量預測論文[3]~[6]中的特徵選擇（即他們的Model Inputs是什麼）。選定數個特徵後開始對每個特徵清理，我們的清理方式分為以下三種：

|  |  |  |
| --- | --- | --- |
| 種類 | 該特徵中的資料 | 清理方式 |
| 第一種 | 缺一點 | 刪除缺少該特徵的那幾筆資料 |
| 第二種 | 缺一些 | 補平均值 |
| 第三種 | 缺很多 | 刪除該特徵 |

如下圖，假設特徵2之中有缺項(NAN)，若為第一、三種情況，其對應刪除方向如圖所示，若為第二種，則把NAN換成特徵2的平均值，分成三種的目的是為了在資料筆數、特徵原始程度、特徵數量之間權衡，我們希望保留一定量的資料筆數，越原始的特徵（而不是幾乎都是填補的平均值）以及盡量多的特徵數量。另外，判別屬於哪一種的臨界值沒有進一步分析和測試，若要證實資料清理的有效性，可以透過輸出乳量的RMSE為指標校正，找出最好的清理方式。

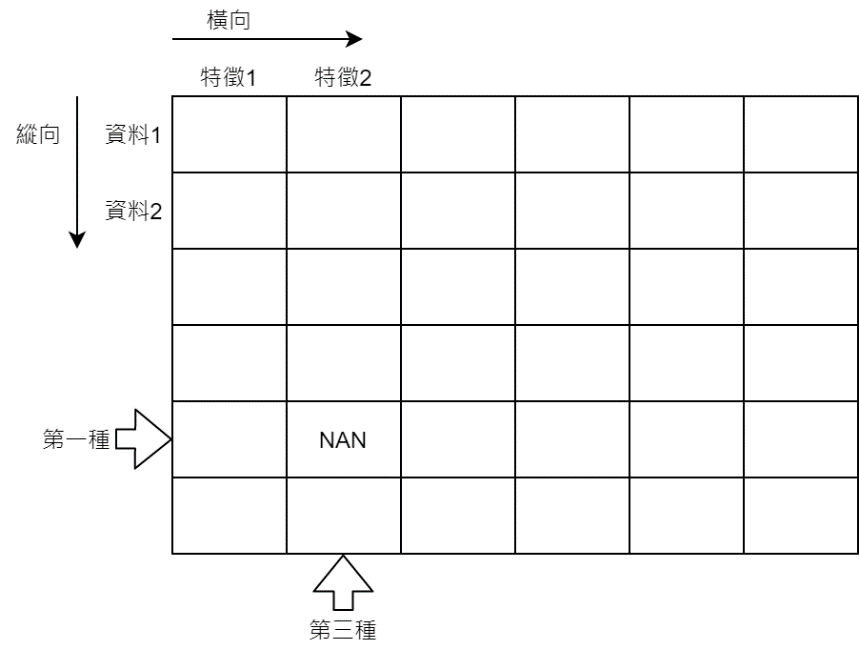


圖3

縱向清理是對資料作清理，我們透過特徵的平均值與標準差為基準判斷資料中的離群值，但我們是針對人為勘誤作刪除，所以並不是將資料視為N維向量，以範數為度量計算其偏離平均值的程度並加以刪除，因為這樣可能不小心刪除到有用的資料，而是一次一個特徵，判斷該特徵中是否有不合理的值（應該是正數但卻是負數、超出好幾個標準差之外），並刪除此筆資料。

* + 1. 資料整合

圖2中的資料整合用於整合資料庫中的特徵及資料，包含生成新的特徵（依月份生成對應該地區的平均氣溫）、特徵間的運算（日期相減生成間隔）以及不同資料庫間的合併（不同資料庫會有不同的紀錄、排序方式，所以需透過相同的特徵為索引合併成相同格式），詳細架構如圖4。

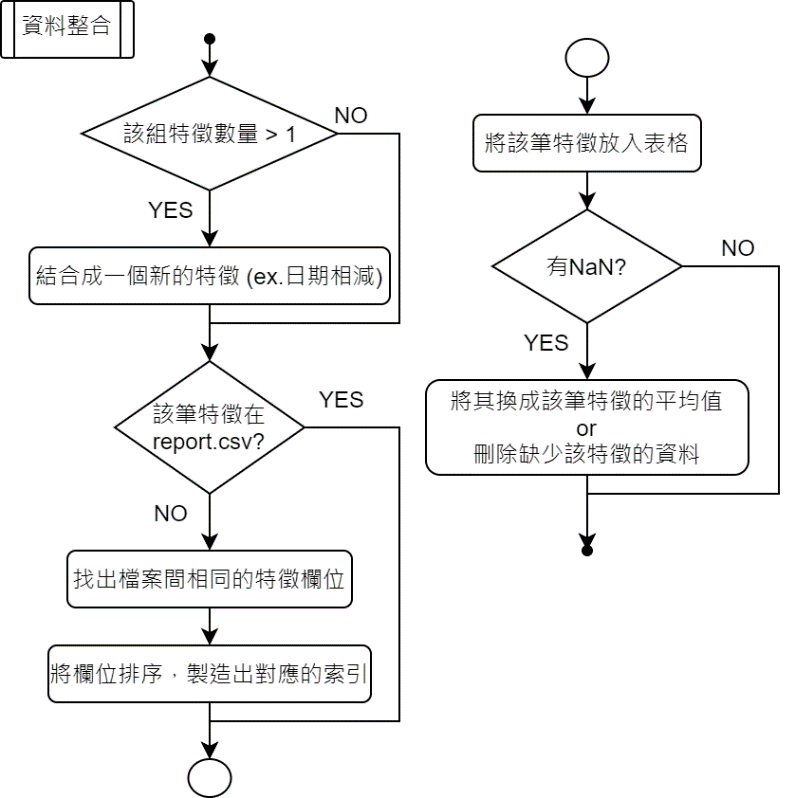


圖4

* + 1. 資料轉換

圖2中的資料轉換架構如下，進行資料轉換時，代表已經由前面的步驟產生出一個含有我們要的所有特徵的表格，此部分會對類別資料作one-hot coding轉換，非類別資料作Z-score正規化。

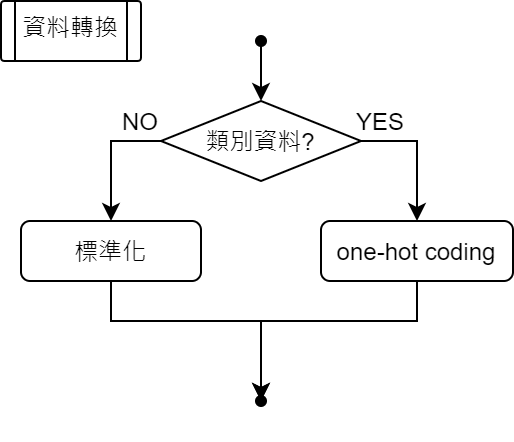


圖5

* + 1. Model Inputs

共33253筆資料、每筆資料8個特徵（胎次、泌乳天數、檢測日期氣候、月齡、配種次數、分娩間隔、乾乳期、農場代號）。

* 1. 機器學習模型
     1. BLR(Bayesian linear regression)

Scikit-learn套件裡有各種回歸法可以運用，我們這次使用這個套件裡的LinearRegression和Ridge函式來實現貝氏回歸。

* + 1. Neural Network

神經網路我們使用Keras套件來建立，層數為2或3層，隱藏層的

維度設128，激發函數使用Relu，batch size及epoch數會根據訓練資

料的收斂速度作調整，誤差計算以均方誤差為基準，優化方式選擇

Adam優化器。

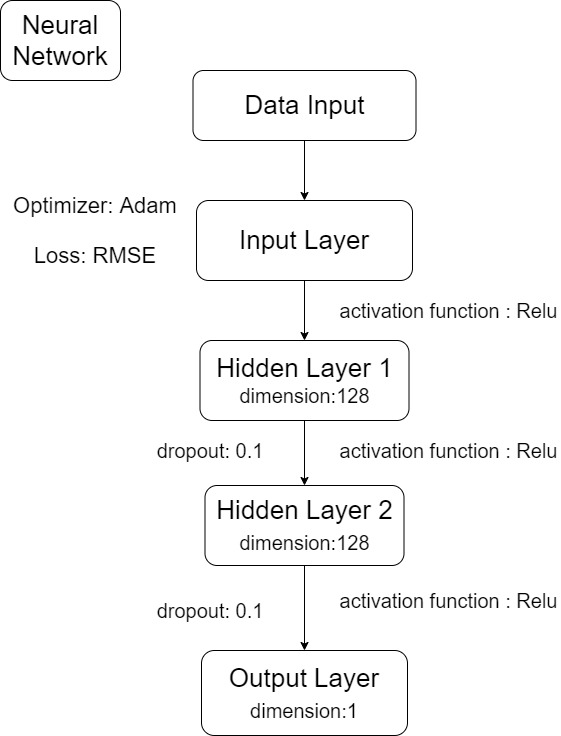


圖6

* + 1. XGBoost (Gradient Boosting Machine)

XGBoost的原理是將多個決策樹跟和Gradient Descending以及Boosting結合在一起，並做一些改良，像是考慮誤差函數考慮二階泰勒展開、損失函數中引入正則化項，Machine的部分使用scikit-learn的API接口來實作，XGboost裡的XGBRegressor來做回歸，參數的部分根據內建的Gridsearch套件來篩選參數。

此外我們也運用XGboost原生接口的回歸來算出每個訓練資料的特徵分數，其特徵重要性的會依據下列三個部分做計算：

* + - 1. 使用特徵在所有樹中作為劃分屬性的次數
      2. 使用特徵在作為劃分屬性時loss平均的降低量
      3. 使用特徵在作為劃分屬性時對樣本的覆蓋度

1. Results
   1. 預測結果(表格內數字為RMSE，最佳private scoreboard)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **資料**  **模型** | **資料有正規化** | | **資料無正規化** | |
| **考慮地點最高/低溫度** | **無考慮地點最高/低溫度** | **考慮地點最高/低溫度** | **無考慮地點最高/低溫度** |
| **BLR** | 6.6463864 | 6.5585554 | 6.7017657 | 6.6614588 |
| **NN** | 6.3896480 | 6.2207780 | 6.0679364 | **5.9699810** |
| **XGBoost** | 6.0136973 | 6.0331024 | 6.0092821 | 6.0301512 |

圖7

* Neural Network與XGBoost Regression相對表現較好
* 資料正規化會加快收斂速度，節省訓練時間，但NN在資料正規化後誤差較大
* 考慮地點最高/低溫反而會增加BLR的誤差
* 考慮地點最高/低溫可略微降低XGBoost訓練出來的誤差
* 適當地增加特徵數，並搭配合適的模型有助於找出更精確的解
  1. 特徵重要性

此外我們用XGBoost工具裡的 Feature Importance 畫出圖表，可以很明顯的看出每個特徵的重要性，觀察哪些因素對乳量的影響是最大的。

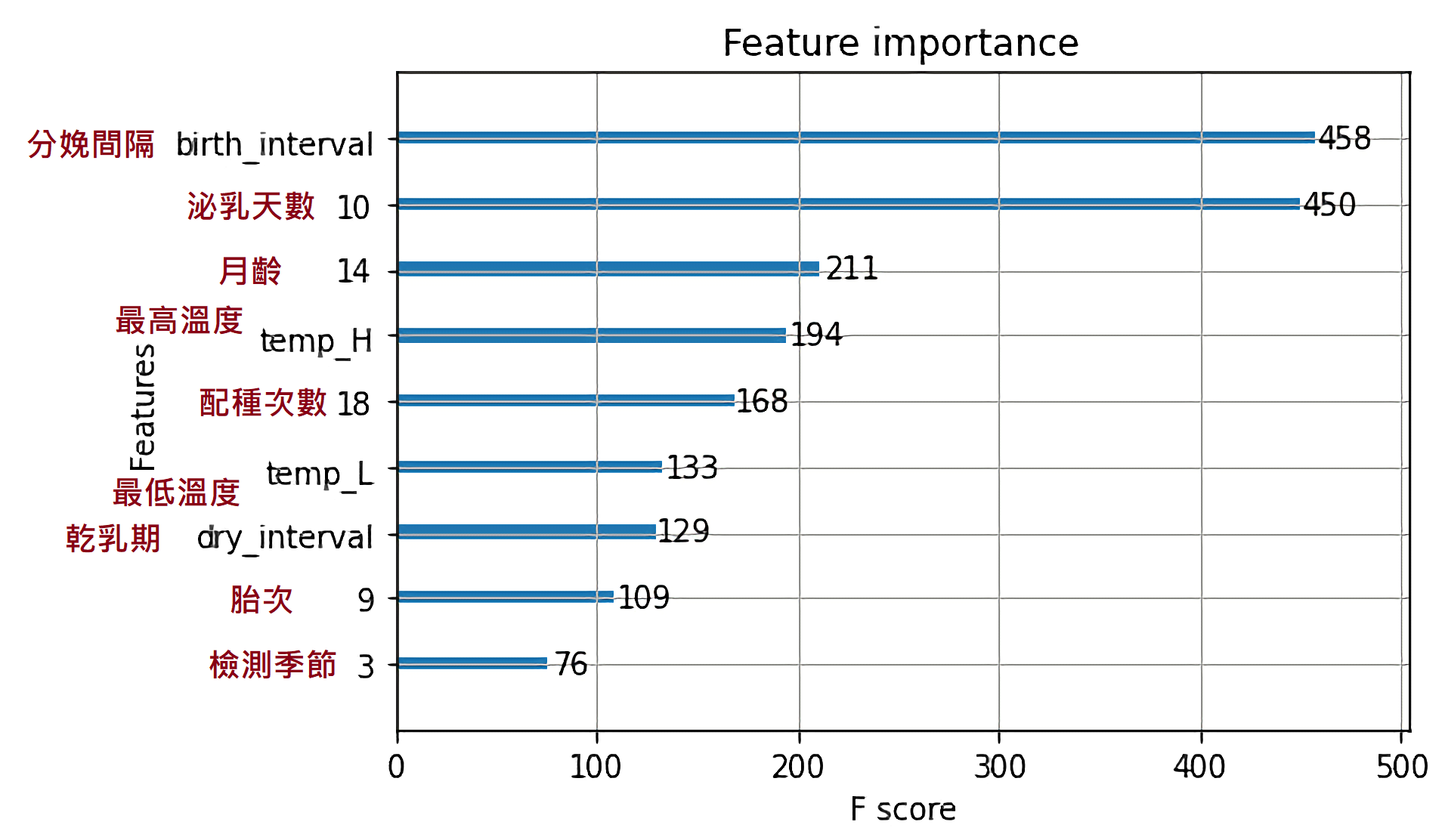


圖8

* 牧場地點影響因素較多，難以列為比較特徵，故沒有計算
* one hot encoding來表示的特徵沒辦法比較，故這邊季節的特徵在計算特徵重要性時使用的是標籤化的特徵
* 可以看出泌乳天數跟分娩間隔是影響乳量的重要因素

1. Summary

在乳量預測的過程中，最重要的是資料的處理，影響乳量的因素非常多，我們從現有的資料中篩選數據，過濾掉不必要的資訊，提取出重要的特徵，最後根據資料的特性選擇訓練模型進行預測，並觀察預測的結果來優化模型，透過不斷的嘗試，從失敗中學習，同時參考相關的論文，才能精進數據分析的能力，以此達到更精確的預測，除了預測乳量之外，我們還可以利用現有的工具找出最容易影響乳量的關鍵特徵，發現泌乳的天數和分娩日間的休養時間與乳量有密切關係，根據這些特徵來調整飼養乳牛的方式，想必對整個酪農產業有所幫助。

V. Reference

1. <https://www.coa.gov.tw/ws.php?id=2501744>
2. <https://medium.com/jameslearningnote/%E8%B3%87%E6%96%99%E5%88%86%E6%9E%90-%E6%A9%9F%E5%99%A8%E5%AD%B8%E7%BF%92-%E7%AC%AC2-4%E8%AC%9B-%E8%B3%87%E6%96%99%E5%89%8D%E8%99%95%E7%90%86-missing-data-one-hot-encoding-feature-scaling-3b70a7839b4a>
3. <https://www.sciencedirect.com/science/article/pii/S0022030214002690>
4. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.626.3829&rep=rep1&type=pdf>
5. <http://www.scielo.org.za/scielo.php?script=sci_arttext&pid=S0375-15892012000300010>
6. <https://www.sciencedirect.com/science/article/pii/S0168169906000998>
7. <https://brohrer.mcknote.com/zh-Hant/using_machine_learning/find_the_right_algorithm.html>
8. <https://machinelearningmastery.com/spot-check-regression-machine-learning-algorithms-python-scikit-learn/>
9. <https://www.itread01.com/elpc.html>
10. <https://xgboost.readthedocs.io/en/latest/>
11. <https://keras.io/zh/>
12. <https://www.kaggle.com/phunter/xgboost-with-gridsearchcv>
13. <https://machinelearningmastery.com/feature-importance-and-feature-selection-with-xgboost-in-python/>