**AI CUP 2024 秋季賽**

附件

**根據區域微氣候資料預測發電量競賽報告**

隊伍：TEAM\_5729

隊員：甯文駿(隊長)

Private leaderboard：554811.0 / Rank 18

**壹、環境**

本機作業系統和主要硬體設備：

* Windows 10 專業版
* CPU：11th Gen Intel(R) Core(TM) i7-11700 @ 2.50GHz
* GPU：NVIDIA GeForce RTX 3080
* RAM：128 GB

使用語言和版本：

* Python：3.9.12

主要套件（函式庫）：

1. 數據處理與分析

* pandas：用於數據處理。
* numpy：提供高效的數值計算和數據處理。
* missingno：用於視覺化和處理缺失值。

2. 模型構建與訓練

* scikit-learn：全面的機器學習工具，包括分類、迴歸、評估等。
* xgboost：一種基於梯度提升框架的樹模型
* catboost：用於處理分類和迴歸問題，支持類別型數據。
* Ridge：一種正則化線性迴歸方法。
* tensorflow/keras：用於深度學習模型的構建與訓練。
* joblib：用於模型的保存和載入模型。

3. 超參數調整

* optuna：強大的超參數調整工具。

4. 資料視覺化

* matplotlib：基本的資料視覺化工具。
* seaborn：資料視覺化工具。

安裝套件內容放在**requirements.txt**裡，pip install -r requirements.txt即可。

額外資料集：

* ADD\_FEATURE.csv

該資料是內容為**競賽與資料格式說明\_V2**裡所提供的資料蒐集裝置位置。使用方式是透過LocationCode將兩個 DataFrame(df和add)進行合併。並將

Latitude(緯度)、Longitude(經度)、Face(面朝方向)、Height(高度)，列入模型的輸入特徵。

表 1\_ADD FEATURE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LocationCode | Latitude | Longitude | Face | Height |
| 1 | 23.8994 | 121.5444 | South | 5 |
| 2 | 23.8997 | 121.5447 | South | 5 |
| 3 | 23.8997 | 121.545 | South | 5 |
| 4 | 23.8994 | 121.5444 | South | 5 |
| 5 | 23.8994 | 121.5447 | Southwest | 5 |
| 6 | 23.8994 | 121.5444 | Southwest | 5 |
| 7 | 23.8994 | 121.5444 | South | 5 |
| 8 | 23.8997 | 121.545 | Southwest | 3 |
| 9 | 23.8994 | 121.5444 | Southeast | 3 |
| 10 | 23.8994 | 121.5444 | Southwest | 1 |
| 11 | 23.8997 | 121.5447 | Southeast | 1 |
| 12 | 23.8997 | 121.5447 | Northwest | 1 |
| 13 | 23.8978 | 121.5394 | West | 5 |
| 14 | 23.8978 | 121.5394 | South | 5 |
| 15 | 24.0092 | 121.6172 | Southeast | 1 |
| 16 | 24.0092 | 121.6172 | East | 1 |
| 17 | 23.9933 | 121.5985 | Northwest | 3 |

LocationCode17為私人住宅不提供座標與高度，只知道位於花蓮市，且距離花蓮市氣象站(花崗街)約2.8公里，因此從Google 地圖上找了一個符合條件的建築物當作參考，可透過該經緯度搜尋該物件。

**貳、演算方法與模型架構**

在這次的比賽當中，我主要是使用機器學習 (Machine Learning)的**集成學習(Ensemble Learning)**模型來做訓練，希望透過多個模型產出更強大的一個模型。使用模型包括：Random Forest Regressor(Bagging)、XGBoost Regressor(Boosting)、CatBoost Regressor(Boosting)、Ridge Regression。

最後，透過**Stacking(堆疊法)**的方式，將部分模型合併在一起避免模型過擬合，透過多個基底的模型讓最終預測結果有較佳的表現。

1. 樹模型

* Random Forest Regressor

隨機森林的演算法是由CART(Classification and regression tree)所建構的，並透過Bagging演算法，在原始數據集上進行隨機抽樣來創建多個子集，然後在每個子集上訓練一個模型，在構造每棵決策樹的過程中，隨機選擇部分特徵來進行分裂，而不是考慮所有特徵，進一步降低模型的相關性。最後，透過對所有模型的輸出進行平均或多數投票來獲得最終結果。

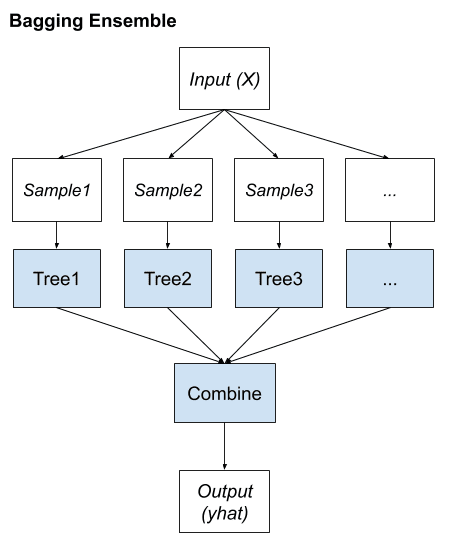


圖 1\_Bagging

(資料來源：[Po-Kai Huang](https://medium.com/@pokaihuang_90287?source=post_page---byline--f575461889bd--------------------------------)，2023，集成學習-Ensemble Learning)

模型參數與超參數：

一張含有 文字, 螢幕擷取畫面, 陳列, 字型 的圖片

自動產生的描述

圖 2\_RF模型

(資料來源：自行製作)

* XGBoost Regressor

XGBoost(Extreme Gradient Boosting)，是採用梯度提升決策樹(GBDT）演算法。Boosting 主要是想透過學習基於之前的錯誤中進步，例如第二個模型會修改第一個模型的預測結果，第三個模型則是修改第二個。Boosting 中每一個模型都是有關聯的，目標是希望訓練後，新的模型在遇到這些較難預測的資料時可以表現得更好。

一張含有 文字, 圖表, 行, 工程製圖 的圖片

自動產生的描述

圖 3\_Boosting

(資料來源：AWS，演算法的運作方式 SageMaker XGBoost)

模型參數與超參數：

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

圖 4\_XGB模型

(資料來源：自行製作)

* CatBoost Regressor

CatBoost(Categorical Boosting)與XGBoost相同，採用梯度提升決策樹(GBDT）演算法，專為解決迴歸、分類和排序問題設計。與傳統梯度提升方法相比，CatBoost 對於類別型特徵處理、速度優化及防止過擬合表現優秀。該演算法的主要目標是提高模型準確性和穩定性，特別適合用於具有類別特徵的大型數據集。

模型參數與超參數：

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

圖 5\_CatBoost模型

(資料來源：自行製作)

1. 線性模型

* Ridge Regression

Ridge Regress(嶺迴歸)是一種線性迴歸方法，在基於普通線性迴歸基礎上添加了 L2 正則化來限制迴歸係數的大小。這個正則化項是係數平方和，L2 正則化會對迴歸係數的平方和進行懲罰，有助於減少模型的複雜度，從而提高模型的穩定性。

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

圖 6\_Ridge模型

(資料來源：自行製作)

1. 堆疊模型

* StackingRegressor

Stacking 一樣是由多個模型組成，但是與 Boosting 和 Bagging 不同的地方是，每個模型可以選擇相同或不同的演算法，訓練完多個模型後，我們要將這些模型合併在一起。合併的方式是另外訓練一個模型，這個模型把前面所有模型的輸出當成輸入，並產生出最終預測結果。

一張含有 文字, 圖表, 行, 螢幕擷取畫面 的圖片

自動產生的描述

圖 7\_Stacking

(資料來源：[Po-Kai Huang](https://medium.com/@pokaihuang_90287?source=post_page---byline--f575461889bd--------------------------------)，2023，集成學習-Ensemble Learning)

**Estimators(基模型)**：XGBoost、 RandomForest、CatBoost

**Final\_estimator(最終模型)**：Ridge迴歸

一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述

圖 8\_Stacking模型

(資料來源：自行製作)

**參、創新性**

本競賽所使用的機器學習模型，充分展現了集成學習(Ensemble learning)在迴歸分析中的創新應用，我結合了隨機森林(Random Forest)、XGBoost和CatBoost等模型的優勢，通過多個基模型的堆疊來有效提升迴歸分析的準確性，最終在測試集中實現了 R2 Score高達0.93的優異表現。

創新之處在於運用了迴歸分析針對Sunlight(Lux)的缺陷進行數值上的修正，結合季節特徵，最後透過相關性熱力圖進行精確的特徵選擇。

模型超參數訓練，使用了Optuna套件，針對模型做更進一步的優化，顯著改善模型的泛化能力，尤其是XGBoost和CatBoost，在測試集中的R2 Score分別達到了0.91和0.92，顯示出集成學習的高精確性。

最終，多模型的堆疊架構結合超參數訓練技術，不僅大幅減少了預測誤差，還為能源管理等複雜應用場景提供了穩定的解決方案，也彰顯了機器學習技術的創新價值。

**肆、資料處理**

個別處理：

1. 讀取資料(L1-L17\_Train.csv)，如有TrainingData\_Additional\_V2，則在讀取該資料時透過pd.concat來做合併處理。
2. 透過describe()做統計描述，查看個欄位的數值分佈是否有不合理的地方(像是L1\_Train.csv有一筆資料的Humidity(%)達到了300)，如有則做刪除處理。

一張含有 文字, 螢幕擷取畫面, 字型, 設計 的圖片

自動產生的描述

圖 9\_L1\_Train.describe()

(資料來源：自行製作)

1. 利用迴歸分析先分析Sunlight(Lux)數值尚未達到最大值前的Sunlight(Lux)對Power(mW)的比率，回推Power(mW)所產生的Sunlight(Lux)。
2. 修改欄位名稱以符合upload.csv的序號格式(序號西元年(4碼)+月(2碼)+日(2碼)+預測時間(4碼)+裝置代號 (2碼)，總共14碼)，處理過程包括將LocationCode轉換為兩位數字的字符串格式、提取日期時間的相關欄位來重組DateTime欄位、10筆資料做平均合成一筆資料、欄位重新命名等。

一張含有 文字, 螢幕擷取畫面 的圖片

自動產生的描述

圖 10\_L1\_Train\_Revise.csv

(資料來源：自行製作)

1. 透過.info()和.describe()來查看資料是否處理成功，最後將處理好的資料合併並輸出成LALL\_Train\_Revise.csv。

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

圖 11\_LALL\_Train\_Revise.csv

(資料來源：自行製作)

總處理：

1. 讀取資料(ADD\_FEATURE.csv)，將類別特徵[‘Face’]做One-Hot編碼，隨後與原始DataFrame合併並刪除舊欄位。

一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述

圖 12\_Face One-Hot 編碼

(資料來源：自行製作)

1. 讀取資料(LALL\_Train\_Revise.csv)，從Serial欄位中切割出日期、時間等資訊。
2. 定義季節劃分函數(3月到5月為Spring，6月到8月為Summer等)，新增['Season']欄位並應用函數，之後對該欄位進行One-Hot編碼。
3. 最後根據LocationCode來合併csv，作為模型訓練所使用的輸入資料。

一張含有 文字, 螢幕擷取畫面, 軟體, 數字 的圖片

自動產生的描述

圖 13\_Training.csv

(資料來源：自行製作)

**伍、訓練方式**

1. 決定模型訓練所使用的特徵

計算與Power(mW)的相關性矩陣並繪製相關性熱力圖，直觀展示不同特徵之間的關聯，以此來作為輸入特徵的選擇參考。

1. 切分資料集

test\_size=0.25：將資料切分成75%用於訓練的訓練集，25%的用於測試的測試集。

random\_state=42：設置隨機種子，這樣每次執行程式碼時，切割方式都會相同，有助於結果的重現。

1. 對訓練集和測試集進行標準化處理

使用StandardScaler進行標準化處理，將特徵轉換為均值為0，標準差為1的分佈。首先對訓練集x\_train做fit\_transform()處理，接著測試集x\_test做transform()，測試集使用的是transform()，而不是fit\_transform()的原因是，測試集應該使用與訓練集相同的標準化參數。

1. 初始模型訓練及評估

選擇幾個迴歸模型，定義模型評估函數，在這裡我使用三個常見的迴歸評估指標(R2 Score、MSE、MAE)，接著使用for迴圈來對每個模型做訓練，每次訓練後，都會列出該模型的訓練和測試結果。

一張含有 文字, 螢幕擷取畫面, 字型, 軟體 的圖片

自動產生的描述

圖 14\_選擇模型

(資料來源：自行製作)

一張含有 文字, 螢幕擷取畫面, 軟體 的圖片

自動產生的描述

圖 15\_評估指標函數與訓練

(資料來源：自行製作)

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

圖 16\_初始模型評估

(資料來源：自行製作)

1. 超餐數調整及評估

使用Optuna套件，針對個別模型做超參數調整，期望得到更好的結果。

一張含有 文字, 螢幕擷取畫面, 軟體, 陳列 的圖片

自動產生的描述

圖 18\_Opruna-XGB

(資料來源：自行製作)

**陸、分析與結論**

所有模型訓練都是使用相同資料集、特徵和相同的訓練測試比例，模型績效則是輸出訓練和測試的R2 Score、MSE(均方誤差)、MAE(平均絕對誤差)來做分析。

**RF模型分析：**

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

圖 19\_RF模型分析

(資料來源：自行製作)

* 訓練

R2 Score：在超參數調整前後，R2 Score均為0.99，表明模型對訓練數據的擬合能力已非常高，且基本沒有變化。

MSE/MAE：超參數調整後，訓練誤差進一步下降，這表明超參數調整使模型在訓練集上更準確。

* 測試

R2 Score：超參數調整前後，測試集 R2 Scor從 0.92 增加到 0.93，有小幅提升。

MSE/MAE：測試誤差有細微下降，表明模型的泛化能力稍微改進。

**XGB模型分析：**

一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

圖 20\_XGB模型分析

(資料來源：自行製作)

* 訓練

R2 Score：在超參數調整前後，R2 Score從0.81提升至0.99。

MSE/MAE：超參數調整後，MSE和MAE大幅下降，表明模型能很好地擬合訓練數據。

* 測試

R2 Score：超參數調整前後，測試集 R2 Scor從 0.8增加到 0.91，提升不少。

MSE/MAE：MSE 和 MAE 大幅下降，表明泛化能力改善明顯。

**CatBoost模型分析：**

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

圖 21\_CatBoost模型分析

(資料來源：自行製作)

* 訓練

R2 Score：在超參數調整前後，R2 Score從0.78提升至 0.99。

MSE/MAE：超參數調整後，MSE和MAE大幅下降，表明模型能很好地擬合訓練數據。

* 測試

R2 Score：超參數調整前後，測試集 R2 Scor從0.77提升至0.92，泛化能力改善明顯。

MSE/MAE：MSE 和 MAE 大幅下降，泛化能力改善明顯。

**Ridge模型分析：**

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

圖 22\_Ridge模型分析

(資料來源：自行製作)

* 訓練

R2 Score：在超參數調整前後，R2 Score從0.05提升至0.51，有顯著改善，但仍屬於中等水平。

MSE/MAE：超參數調整後，MSE和MAE大幅下降。

* 測試

R2 Score：超參數調整前後，測試集 R2 Scor從0.05提升至0.5，仍屬於中等水平。

MSE/MAE：MSE 和 MAE 大幅下降，測試集上的預測誤差顯著降低。

**Stacking模型分析：**

一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

圖 23\_Stacking模型分析

(資料來源：自行製作)

* 訓練

R2 Score：訓練集R2 Score為0.98，表明堆疊模型在訓練數據上的解釋能力非常強。

MSE/MAE：MSE 為 3822.58，MAE 為 25.78，顯示模型對訓練數據的預測誤差非常低。

* 測試

R2 Score：訓練集R2 Score為0.93，泛化能力優秀。

MSE/MAE：MSE為14467.43，MAE為53.77，表明模型的預測誤差在測試數據上維持在較低水準，性能穩定。

**結論：**

* **模型方面：**

堆疊模型裡結合了多種迴歸器，並使用多項式迴歸作為最終層迴歸器，所帶來的成果就是R2 Score、MSE和MAE與單一模型相比皆有更好的表現，但訓練時間卻達到了143分18秒，反映出多個基礎模型(如XGBoost和CatBoost的多迭代數)和多項式迴歸增加了計算成本，之後可嘗試減少基礎模型數量、降低多項式階數，或在基礎模型中調整參數以縮短訓練時間，或是改用Deep Learning深度學習模型(如LSTM和BiLSTM)來做訓練，亦或是從源頭的輸入資料來做修改。

* **資料方面：**

模型訓練所輸入的資料皆為比賽官方所提供，無使用其他外部資料，如果加入了公開的第三方資料(如氣象資料開放平台、中央氣象署的氣象資料等)，模型的績效和訓練時間一定會改善許多，也能增加運用AI的能力和對太陽能的認識。

**柒、程式碼**

Github連結：<https://github.com/c110156247/AI-CUP-2024-Competition-to-predict-power-generation-based-on-regional-microclimate-data/tree/main>

**捌、使用的外部資源與參考文獻**

1. 楊智堯. (2024, February 29). *集成學習（Ensemble Learning）*. HackMD. <https://hackmd.io/@OJo2ruXGShKdpuewtwzZcQ/B1gcip3n6>
2. 黃柏愷, & 鄭雅綿. (2023, May 24). *集成學習-Ensemble Learning*. Medium. <https://medium.com/data-science-navigator/0%E9%9B%86%E6%88%90%E5%AD%B8%E7%BF%92-ensemble-learning-f575461889bd>
3. *演算法的運作方式 SageMaker XGBoost*. (n.d.). AWS. <https://docs.aws.amazon.com/zh_tw/sagemaker/latest/dg/xgboost-HowItWorks.html>

**作者聯絡資料表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 隊伍名稱 | TEAM\_5729 | Private Leaderboard 成績 | 554811.0 | Private Leaderboard 名次 | 18 |
| 隊長 | 甯文駿  Wen－Jun,Ning | 國立高雄科技大學智慧商務系 | National Kaohsiung University of Science and Technology Department of Intelligent Commerce | 0965-102-986 | mmbrian6919＠gmail.com |

★註1：請確認上述資料與AI CUP報名系統中填寫之內容相同。自2023年起，獎狀製作將依據報名系統中填寫內容為準，有特殊狀況需修正者，請主動於報告繳交期限內來信moe.ai.ncu@gmail.com。報告繳交截止時間後將不予修改。

★註2：繳交程式碼檔案與報告，請Email至：ailabailab5051@gmail.com，並同時副本至：t\_brain@trendmicro.com與moe.ai.ncu@gmail.com。缺一不可。