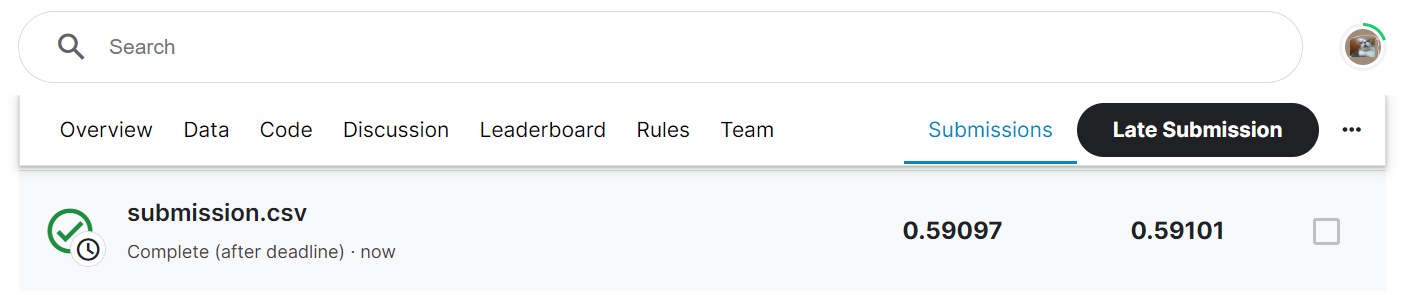
**Final 109550119 邵筱庭**

**Model weight**

<https://drive.google.com/drive/folders/11QsOMRZtqqbmhv_44RS3scFosfWukHHK?usp=sharing>

Score：0.59097, 0.59101



**Environment**

python==3.10.4

numpy==1.23.3

pandas==1.5.1

IPython== 8.5.0

matplotlib==3.5.2

seaborn==0.12.2

colorama==0.4.4

sklearn==0.0

lightgbm==3.3.4

scipy==1.8.1

imblearn==0.0

**Implementation details**

* Outline

1. Problem Introduction
2. Data Processing
3. Model Architecture
4. Code
5. Problem Introduction

在這次的題目中，主要在解決資料問題

1. 如何處理缺失資料

2. 觀察資料間關係

i. 找出與答案關聯度高的features

ii. 組合有關連性的features

1. Data Processing

1. New feature (沒有全部用到，但若要改寫則可以嘗試使用這些 feature)

根據 Kaggle 上的討論，新增多個新的欄位

a. m3\_missing, m5\_missing：當 measurement\_3, measurement\_5 為空，missing = 1，否則為0

b. m3\_17\_avg：為measurement\_3 和measurement\_17的平均值

c. m3\_17\_stdev：為measurement\_3 和measurement\_17的標準差

d. m3\_to\_16\_avg：為measurement\_3 至measurement\_16的平均值

e. area：attribute\_2乘上attribute\_3

2. Data impute

使用HuberRegressor和KNNImputer來補齊欄位內的值

3. SMOTE (最終沒有使用)

由於label 0和1的比例不均，因此可考慮使用SMOTE處理資料不平衡的問題。

結果：一開始獲得較高的分數，但在固定random\_state後，分數反而更低了。由於較高分數與使用SMOTE差距不大，因此選擇較穩定的方法(不使用SMOTE)

4. Reference

新增與處理feature：

* + - 1. <https://www.kaggle.com/competitions/tabular-playground-series-aug-2022/discussion/343939>
      2. <https://www.kaggle.com/competitions/tabular-playground-series-aug-2022/discussion/342319>
      3. <https://www.kaggle.com/competitions/tabular-playground-series-aug-2022/discussion/342126>

1. Model Architecture

LogisticRegression：

Params:

max\_iter=1000, C=0.0001, penalty='l2', solver='newton-cg'

**max\_iter**：Maximum number of iterations taken for the solvers to converge

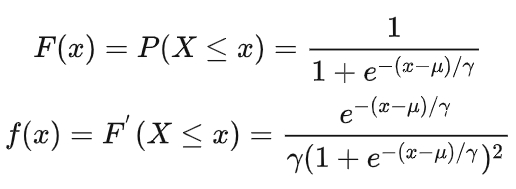
**C**: Inverse of regularization strength; must be a positive float. Like in support vector machines, smaller values specify stronger regularization.

**penalty**: Specify the norm of the penalty ('l2': add a L2 penalty term and it is the default choice)

**solver** : Algorithm to use in the optimization problem (newton-cg：利用Hessian matrix 來迭代以優化損失函數)

Intro:

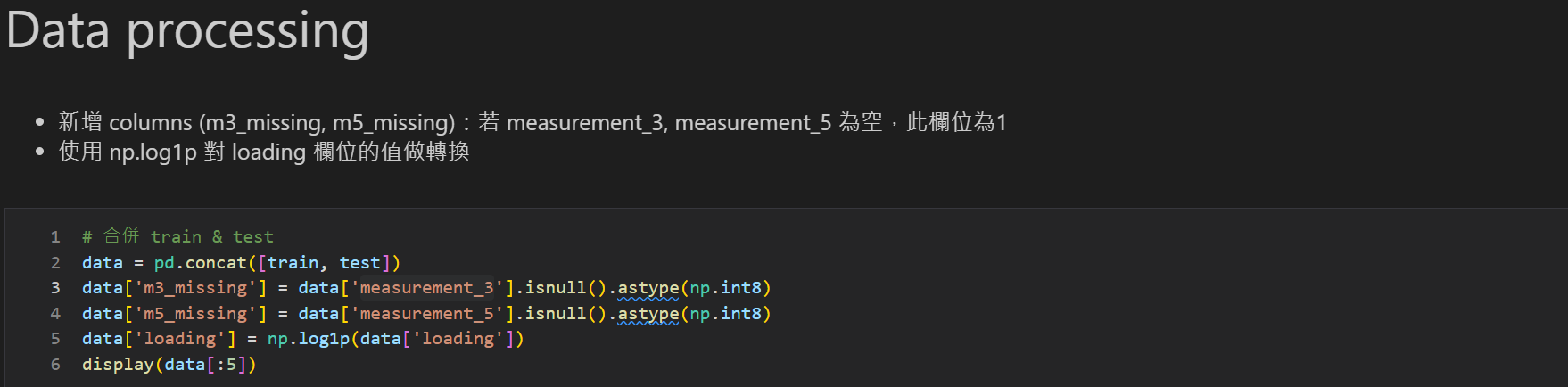
為連續型的機率分佈，有分佈函數及密度函數

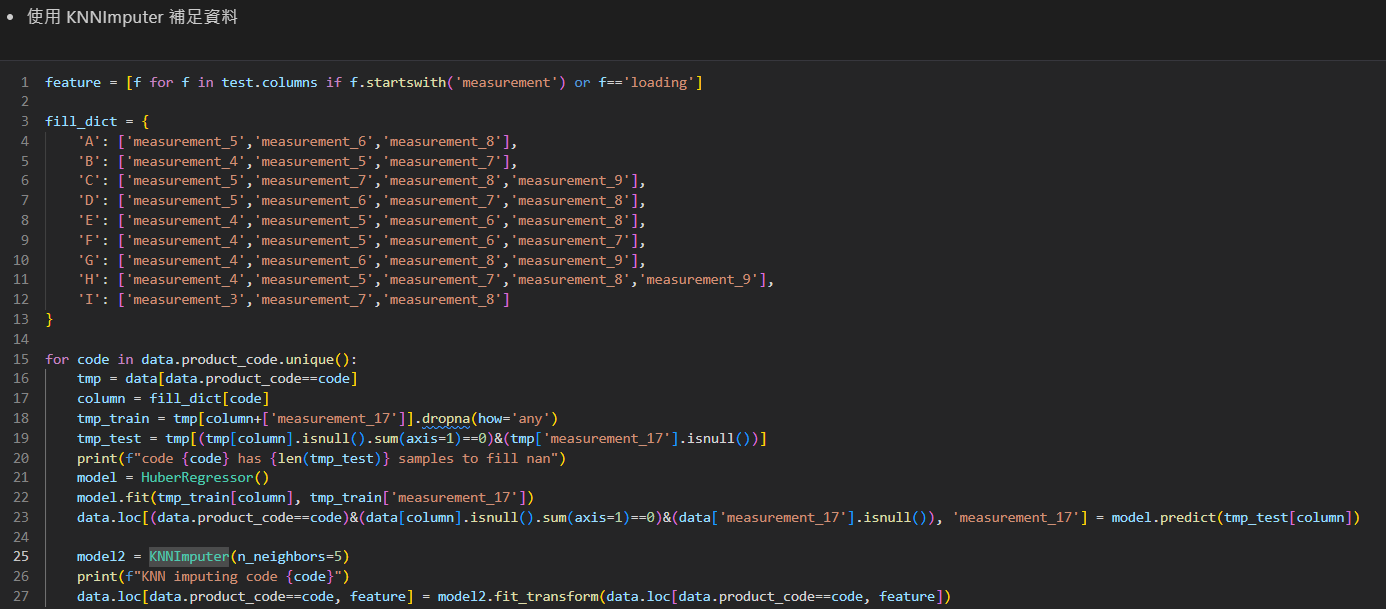
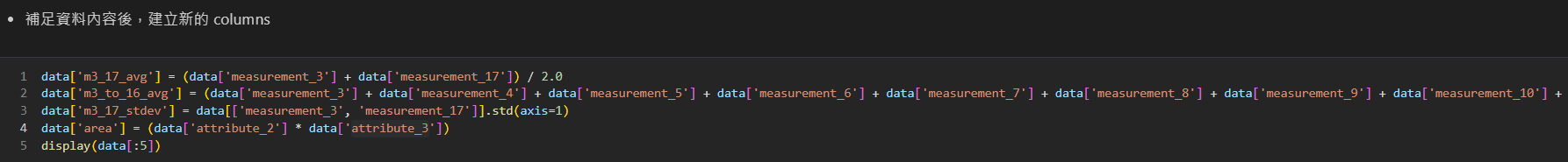


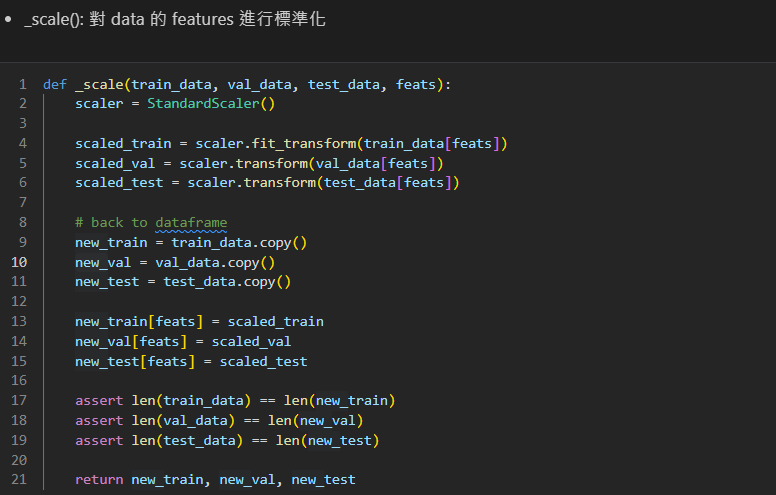
產生出決策邊界，輸入值後根據輸出(是否 > 0 )判斷類別

1. Code

1. Data Process





2. Train

a. 選擇features

根據kaggle上的討論，得知資料間關聯

1. loading、measurement\_17和label關聯性大

2. measurement\_3、measurement\_5有無缺失和label關聯性大

3. 不要一次選取過多feature，效果反而不佳

總結：為了讓許多feature都能影響答案，且不要一次性選取過多feature，因此採取三種features組合，最後在設置權重綜合出答案

Features 組合：

1. select\_feature = ['m3\_missing', 'm5\_missing', 'measurement\_1', measurement\_2', 'loading', 'measurement\_17']

2. select\_feature = ['measurement\_1', 'measurement\_2', 'loading', 'measurement\_17']

3. select\_feature = ['loading', 'measurement\_17', 'm3\_17\_avg']

b. 使用KFold

使用KFold以及LogisticRegression Model，並且計算出使用的feature的重要程度

# lr\_oof\_1：LogisticRegression 在每個驗證集上預測的預測機率。

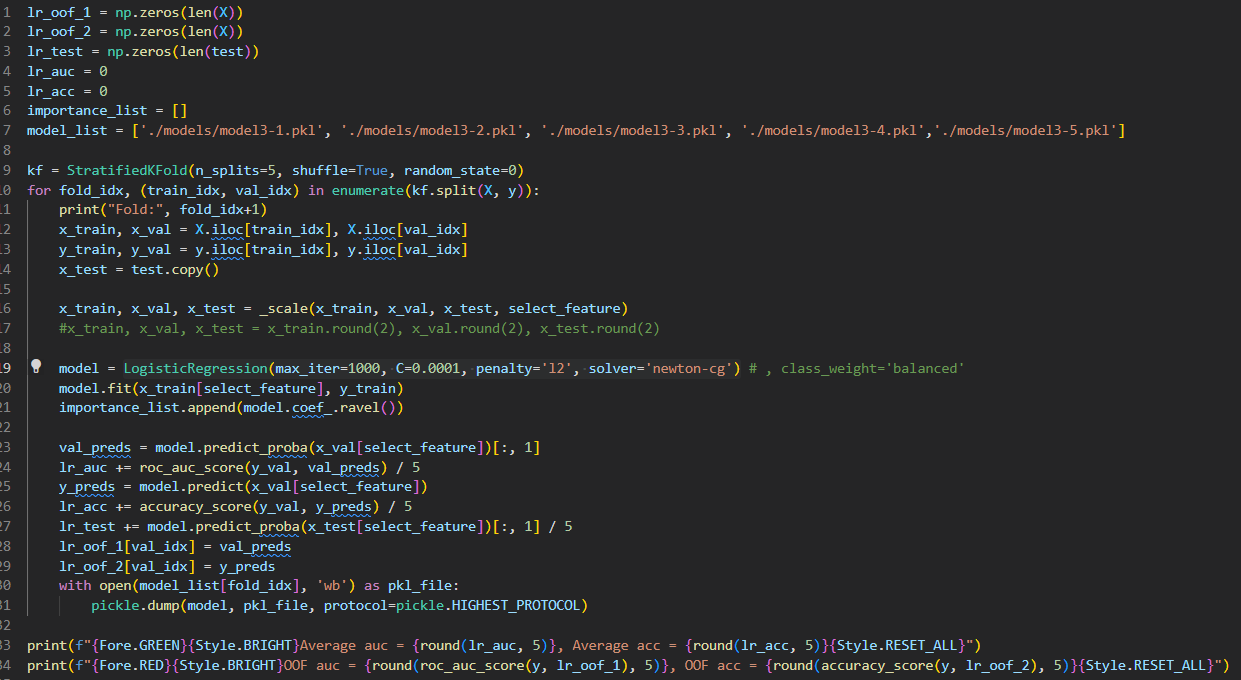
# lr\_oof\_2：LogisticRegression 在每個驗證集上預測的預測類別。

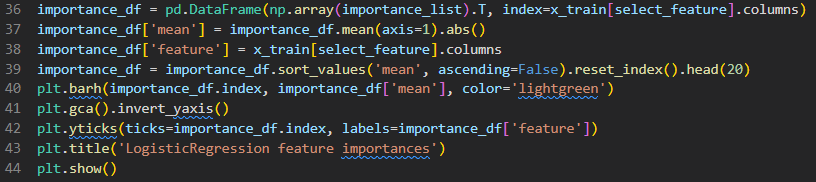
# lr\_test：LogisticRegression 在測試資料集上預測的預測機率。

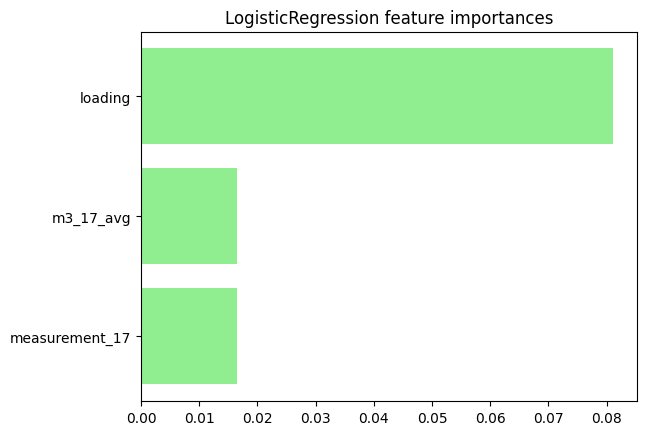
# lr\_auc：LogisticRegression 在交叉驗證過程中的平均 AUC 值。

# lr\_acc：LogisticRegression 在交叉驗證過程中的平均準確度。

# importance\_list：LogisticRegression 在各個 fold 上的特徵重要性。







將使用rankdata，將三個組的lr\_test變成該組合對於label的預測結果

並以權重(0.7, 0.05, 0.3)得到預測結果

