

# NCTU Introduction to Machine Learning, Final

109550018 郭昀

## GitHub Link

[https://github.com/Mars3397/2022\\_Machine\\_Learning/tree/main/Final\\_Project](https://github.com/Mars3397/2022_Machine_Learning/tree/main/Final_Project)  
[Model weight](#)

## Reference

<https://www.kaggle.com/code/ambrosm/tpsaug22-eda-which-makes-sense>

- GroupKFold
- Feature engineering of measurement\_3 & measurement\_5
- Clipping of measurement\_2

## Brief Introduction

本次作業之 task 為根據產品之各項測量結果，預測此產品之 failure。我選擇 logistic regression 當作我的 model，然後另外做了 feature engineering, measurements standardization 以及 fill in missing values 等 data pre-processing, training 的過程則是使用 GroupKFold 來切分 train set 跟 validation set, train 完後將 model 以 pickle 的方式存起來，infernece 的時候即可 load 來使用。

## Methodology

### Data Pre-Process

- LabelEncoder: 因為 product\_code, attribute\_0 以及 attribute\_1 為 string 型態的資料，所以需要將其轉換為數字型態才可以套入 model 中，因此我使用 sklearn.preprocessing 中的 LabelEncoder 來做轉換。
- Standardization: 這部分是我在嘗試提高 performance 的過程中產生的想法，我將 dataset 中的 measurement\_3 ~ measurement\_17 都做 standardization，而結果也有如預期提高，因此我就保留下來了。

- Measurement\_2 clipping: 這部分是在 reference 的討論文章中有提到, measurement\_2 的數值在大於 11 之後, 就會與 failure 成正相關, 因此將 measurement\_2 小於 11 的數值都改為 11。

### Feature Engineering

- Measurement missing: 這部分是參考上方 reference 連結以及[此篇討論之文章](#), 他們都有提出 missing values 也可能造成產品 failure 這個想法, 而根據實驗及統計結果, measurement\_3 跟 measurement\_5 的 missing value count 的 z-score 是所有 features 中的極值, 因此我在 training set 和 testing set 都多加入了 measurement\_3\_missing 和 measurement\_5\_missing 這兩個 feature。
- Average: 這部分則是參考[此篇文章](#), 作者提到可以嘗試將 features aggregate 起來得到有用的 feature, 因此我將 measurement\_3 ~ measurement\_16 全部相加取平均, 並存成 feature “avg”。
- Area: 這個 feature 也是[此篇](#)作者的觀察結果, 作者表示 attribute\_2 和 attribute\_3 看起來像是產品的長跟寬, 因此我將他們相乘並存成 feature “area”

### Fill in missing value

- KNNImputer: 因為 training data 跟 testing data 都含有 Nan, 因此需要將 missing value 補上才能傳入 model 中。我使用的方法是 KNNImputer(n\_neighbors=3) 他會取前後三個值的平均來補 missing value。

### Model Architecture

- Logistic Regression: model 的部分我是選用 LogisticRegression(penalty='l1', C=0.01, solver='liblinear', random\_state=1), 會選用他的原因是我覺得這次的 task 最能提升 performance 的最主要差異會是資料的處理而不是 model 的選擇, 所以就採用比較簡單而且討論區很多人推薦的 LogisticRegression。

### Feature Selection

- 除了於 feature engineering 所述新增加的 feature 外, 我的 feature 還選了 loading, measurement\_17, measurement\_0, measurement\_1, measurement\_2, attribute\_0, 這部分則是我的實驗結果, 嘗試多種組合後得出的最好組合。

### Summary

總結來說，我覺得對於這次的 task 來說，資料的處理非常重要，像是 data pre-processing, feature engineering, fill in missing value 都對於 performance 有顯著的提升。

## Result

**Tabular Playground Series - Aug 2022**  
Practice your ML skills on this approachable dataset!  
Kaggle · 1,888 teams · 4 months ago

OverviewDataCodeDiscussionLeaderboardRulesTeamSubmissionsLate Submission...




### Submissions

You selected 0 of 2 submissions to be evaluated for your final leaderboard score. Since you selected less than 2 submission, Kaggle auto-selected up to 2 submissions from among your public best-scoring unselected submissions for evaluation. The evaluated submission with the best Private Score is used for your final score.

0/2

Submissions evaluated for final score

AllSuccessfulSelectedErrorsRecent

Submission and Description	Private Score ⓘ	Public Score ⓘ	Selected
 <b>109550018.csv</b> Complete (after deadline) · now	0.59046	0.58503	<input type="checkbox"/>
 <b>submission.csv</b> Complete (after deadline) · 1h ago	0.59046	0.58503	<input type="checkbox"/>
 <b>submission.csv</b>	0.59046	0.58503	<input type="checkbox"/>