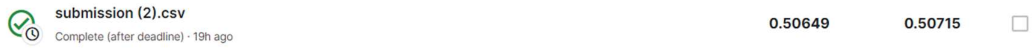
1. Github Link: https://github.com/trainer8888/NYCU-NCTU-ML-Final_Project

2. Reference if you used any code from other resources:

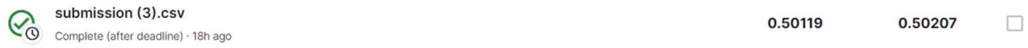
https://www.kaggle.com/code/act18l/stacked-model-mlp-logisticregression-random?scriptVersionId=104560526

3. I use RobustScaler and LogisticRegression to train the model. I also found that data pre-process is VERY important. If I don't pre-process the data, my accuracy always about 0.50.
4. Data pre-process:
   I use WOEEncoder to encode attribute_0 because some discusssions found that attribute_0 is important, but it is string. Then, I log transform the "loading" data, because I found that "loading" and "measurement_17" are the first two important feature after I use SelectKBest(chi2, k=2) to select two best feature. I drop the attribute_1 because it is string. It is hard to fit and not that important. Finally, I use dataframe[feature].isnull().values.any() to check each column whether there are nan data. If so, fill it with median of that column. After I preprocess the data, I use product_code to split the data. For example, product_code A are used for validation and others are used for training.
   Model architecture:
   I use RobustScaler and LogisticRegression to train the model. I also use make_pipeline to make the code shorter. Before I train the model,I drop the product code because it is string, it cannot fit the model. After I fit the model, I fill test["failure"] with 0 because test.csv don't have failure. I just want to fit the training dataframe shape.
   Finally, dump the model and download it. Upload to dataset and load it in the inference code. Use inference code to write submission csv.
5. This final project make me learn more about the data pre-process skill. It is useful when I participate in other machine learning competition.
6. Experimental results: The importance of data pre-process
   Decisiontree **NO** data pre-process (just drop attribute_0, attribute_0, and fill nan)



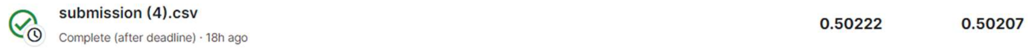
submission (2).csv
Complete (after deadline) · 19h ago
0.50649 0.50715 □

Adaboost (n_estimation = 100) and **NO** data pre-process



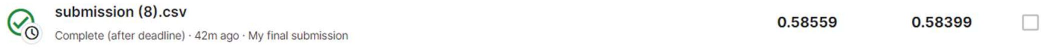
submission (3).csv
Complete (after deadline) · 18h ago
0.50119 0.50207 □

StandardScaler and LogisticRegression and **NO** data pre-process



submission (4).csv
Complete (after deadline) · 18h ago
0.50222 0.50207

This time I encode attribute_0 (also drop attribute_1), log transform the "loading" data, and fill nan.
StandardScaler and LogisticRegression and **YES** data pre-process



submission (8).csv
Complete (after deadline) · 42m ago · My final submission

0.58559 0.58399 □

The accuracy dramatically increased with almost 8%.

Model weight Link:

Kaggle Link
https://kaggle.com/datasets/803d520ff8269425466da14da7edc5e4857b2eb56443138217ae38a8bc9d839c

Google Link
https://drive.google.com/drive/folders/1GfvYkrvZhNuwUJh3Dt84TlFRAMzxMfES?usp=sharing