

Machine Learning Final Project Report

◆ GitHub link of my code:

<https://github.com/wryyy0327/Machine-Learning-Final-Project.git>

◆ Environment details:

Python version: 3.9.7

scikit-learn version: 0.24.2

pandas version: 1.3.4

◆ Introduction:

In this project, we have to analyze the given testing study then construct a model, which can be used to predict the product failure.

To better accomplish this task, we need to choose an appropriate model, analyze the effect of different parameters on the results, and then impute the missing block in the data set.

◆ Methodology:

➤ Data pre-process:

To enhance the model performance, I did the following tricks:

1. It was observed that the loss of data and test results are correlated.
Before impute the missing block, I add a column named “m_5_missing”, which recode that this row miss the measurement_5 data or not.
2. To impute the miss data, I use Sample Imputer with “most_frequent” strategy to do the job. After my testing, this method can impute the data faster and more precise than KNN Imputer or Iterative Imputer.
3. I found that in attribute column, material_5 and material_7 have the greatest influence, so that I encoded them and removed the information of other types of materials.

➤ Model architecture:

To tackle this regression problem, I tried difference ensemble models such as Random Forest or AdaBoost. Finally, I chose Logistic Regression as my model architecture.

➤ **Hyperparameters:**

Model: Logistic Regression with L2 penalty, $C = 0.000005$, solver = 'saga', random state = 0.


➤ **Features selected manually:**

After many times of my testing, I found out that not all of data are helpful for the model to enhance the predict accuracy, so I manually choice following features to fit the model: 'loading', 'material_5', 'material_7', 'measurement_2', 'measurement_10', 'measurement_17', 'm_5_missing'.

◆ **Summery:**

In this project, I consider that the most important point to improve the improve the performance of model is feature selection.

To deal with real world data set, which include many information may not be relevant to our prediction goals, so that I spent most of time to try many difference combinations of features to train my model or even adding some columns to make it better fit the data.

Submission and Description		Private Score ⓘ	Public Score ⓘ
 submission.csv Complete (after deadline) · 3d ago		0.59228	0.58191

After the above operation, the test prediction of my model get 0.59228 in Private Score.

◆ **Code reference:**

I got some ideas from this article on Kaggle discussion:

<https://www.kaggle.com/code/ambrosmp/tps-aug22-eda-which-makes-sense>