109550182 莊婕妤

ML Final Project Report

Brief Introduction

The purpose of this assignment is to predict individual product failures with their test results with past product testing results. I put my main focus on different feature engineering methods and chose to utilize the logistic regression model for the task. The implementation details are listed below.

Refer to my code, the reproduction steps and the pretrained model from :
Github : https://github.com/yu-niverse/Intro-to-Machine-Learning/tree/main/Final_Project
Model Weights : https://reurl.cc/OENqDA

Methodology

I. Feature Engineering Methods

1. Missing Value Imputation
   According to the analysis done by @ambrosm, about 10% of the float feature columns have missing values. I completed the missing values by using k-Nearest-Neighbor imputation (n_neighbors = 15).

2. Feature Construction - Indicate Missing Values of measurement_3 and measurement_5
   Inspired by the discussion posted by @ambrosm, the failure rates of missing measurement_3 and missing measurement_5 deviate significantly from the average failure rate. Thus, I added indicators for the missing_measurement_3 and missing_measurement_5 to the model.

3. Feature Transformation - Label Encoding
   I encoded the string features, product_code, attribute_0 and attribute_1 via label encoding in order to let the model recognize those features. After many experiments, I only take the feature, attribute_0, into account while training.

4. Feature Interaction
   I constructed a new feature, area, which is the product of attribute_2 and attribute_3 inspired by the observation by @desalegngeb.

5. Feature Combination
   I combined features measurement_3 to measurement_17, calculated their average and standard deviation, and added them as two new features while training. (also inspired by the observation by @desalegngeb)

6. Feature Selection
   After many trials, I decided not to take all the attributes into account since some of them might be noises to the model. The chosen attributes include : 'loading', 'measurement_0', 'measurement_1', 'measurement_2', 'measurement_17', 'attribute_0' together with the new attributes I constructed as mentioned above.

7. Standardization
   After applying all the feature engineering methods mentioned above, I standardized all the columns that I took into consideration and found a significant increase in my model performance.

II. Model Selection - Logistic Regression

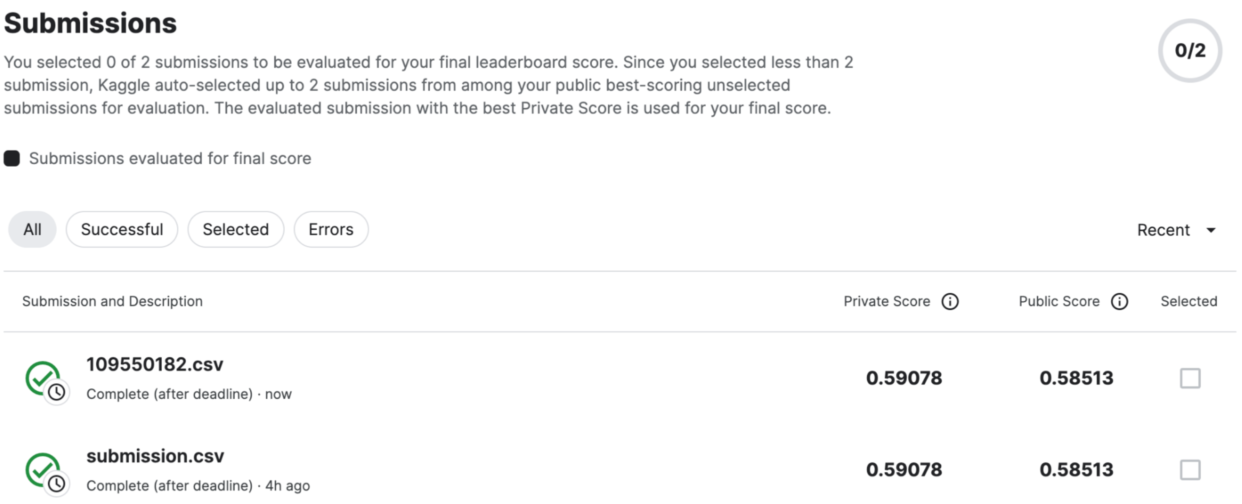
I chose to utilize the logistic regression model since I tried to keep my model simple and put my main focus on preprocessing the attributes. Also, the logistic regression model is recommended by many participants on Kaggle and gives a better performance in comparison to other models including random forest and extra tree classifier according to my experiment.

The parameters I passed in are : penalty='l1', C=0.01, solver='liblinear', random_state=1, referring to the analysis done by @ambrosm.

III. Cross Validation

To check the performance of my model, I did cross validation through group Kfold where the groups are indicated by the product code. Since the test data only includes unseen product codes, performing group Kfold cross validation can simulate the situation. However, I got better results while not using any cross validation techniques at all so I only leverage this method to check my performance.

Results



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

Submission and Description	Private Score	Public Score	Selected
109550182.csv Complete (after deadline) · now	0.59078	0.58513	☐
submission.csv Complete (after deadline) · 4h ago	0.59078	0.58513	☐

Summary

In summary, as inspired by several discussions on Kaggle (listed in reference below), I found out that the major factor that gives good prediction is how you preprocess the given features instead of utilizing a complicated model. Therefore, I tried many feature engineering techniques, filtered out the ones that worked well and resulted in 0.59078 (estimated with AUC).

Reference

- TPSAUG22 EDA which makes sense
- Getting cross-validation right: product codes and GroupKFold
- 17th Place Solution
- Missing values have predictive value
- Less can be more: Feature Engineering Ideas
- A quick summary: dos and don'ts