Machine Learning Final Report

林家輝

109550201

I. Brief Introduction

The company has recently completed an in-depth evaluation of various prototypes for their products and we are now tasked to utilize the dataset obtained from this study in order to anticipate which products will be prone to failure. The data will be employed to build a model that could forecast which products are at risk of not meeting the standards and requirements set by the company. We will be responsible for analyzing the provided dataset and extracting insights that can be used to identify patterns and trends that could indicate a potential product failure. To do this, I tried a wide range of feature engineering and also several different models to predict the product failure.

II. Methodology

DATA PREPROCESSING

I added two features to the dataset which is the count of missing measurement 3 and the count of missing measurement 5. Based on Ambrosm's discussion, The conditional failure rates of both measurements deviate significantly from the average failure rate hence making both a suitable candidate as the additional feature of my prediction model.

Then, I find that there are missing values on most of the measurement features and also loading feature. To deal with this, I utilize KNNImputer which uses the average of the k closest samples to fill in the missing values. I tried various number of neighbour and find that 15 neighbours have the best outcome. I also test other kind of method such as Iterativeimputer and simpleimputer but both of these methods yield a lower accuracy than the KNNImputer.

Next, I encoded the string features (product_code, attribute_0, and attribute_1) with label encoding so that the model can include these features when its training and testing. After attempting to select a specific set of features to include in my training, I ultimately determined that utilizing all available features resulted in the most accurate predictions for my model.

Finally, based on Ambrosm's discussion, there is a correlation between measurement_2 with the target for values above 10. Hence, I clipped this feature value which is less then 11 to 11, in an attempt to improve the accuracy of my model. for my model, I tried clipping only the test dataset and set the train dataset as is. This resulted in a more accurate prediction than clipping both datasets.

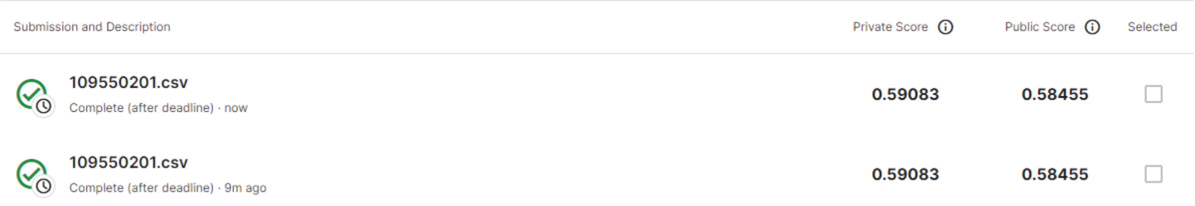
THE MODEL

I Used Logistic Regression as my model due to it being the most accurate model out of the models that I tested and there are a lot of participants that recommends using this model as it also yields the best on their testing.

To find the best hyperparameter, I create a logistic regression model and set it's hyperparameters to a range of numbers. I then run a random search on those hyperparameters to find the best combination that performs well on the data. From my testing, I got (penalty='l1', class_weight='balanced', C=0.01, solver='liblinear') as the best hyperparameter that makes the most accurate model. To increase the accuracy further, I pipelined the model with StandardScaler to standardize the data into the same range and scale.

III. Results and Links

A. Final Private Accuracy



Submission and Description	Private Score ⓘ	Public Score ⓘ	Selected
109550201.csv Complete (after deadline) · now	0.59083	0.58455	□
109550201.csv Complete (after deadline) · 9m ago	0.59083	0.58455	□

B. Github Link

https://github.com/Zryxion/Final-Project-ML

C. My Model Weight

https://drive.google.com/file/d/1Ipo-RpvT137XAkd2t6L-D4exNqvx6jUT/view?usp=sharing

IV. Summary

In summary, I was tasked with building a model to predict product failures using a dataset from a recent study. I first pre-processed the data by adding new features for missing measurements, filling in missing values using KNNImputer, encoding string features with label encoding and clip measurement 2 above 11. Then, I used Logistic Regression model with hyperparameters (penalty='l1', class_weight='balanced', C=0.01, solver='liblinear') as the best combination that performs well on the data. I also pipelined the model with StandardScaler to improve the accuracy even further. One personal lesson that can be learned from this is the importance of data pre-processing, feature engineering, and the impact of different pre-processing techniques to the final results of model. It also shows how proper hyperparameters tuning can bring a substantial difference in the accuracy and performance of the model.