

# Final project report

109550136 邱弘竣

## Brief introduction

The purpose of this project is to predict the failure of Super Soaker made by Keep It Dry. The dataset includes information on measurement, attribute, and product code. These numbers are the results of many experiments. To increase accuracy, we must train our model using these data.

## Methodology

### 1. Set flags of missing values

I create two columns which are "m3\_missing" and "m5\_missing" in both testing data and training data. If the measurement3 is missing, m3\_missing would be filled with value 0, if not, then 1. The same idea is applied to measurement5. The purpose of the flags is that the failure is highly correlated to missing of measurement3 or measurement5 so that I implement this method to record whether it is missing.

### 2. Drop the columns that are useless

I discovered that some columns are modestly correlated to failure. To avoid the model from being distracted by those features, I drop those columns. For example, product code, attribute, etc.

### 3. Fill the missing values

We must predict the values that are empty and fill them in the blank. The method is Iterative Imputer. One thing that we should notice is that measurement17 is highly correlated to failure, therefore, the more accurately we predict the missing values, the higher accuracy we could get. Besides, someone notices that measurement17 is linear combination of measurement3 to measurement9. We could predict the missing value by Linear Regression and find the weights among them.

### 4. Mean

Measurement3 to measurement9 are modestly correlated to failure. Thus, we don't need to train all those columns. What I did is to get their mean value and incorporate them into one column.

## 5. Area

Some participants notice that  $\text{attribute2} * \text{attribute3}$  represents the area of the soaker. As a result, I incorporate into one column.

## 6. Standardization

# Model architecture & Hyperparameters

Logistic Regression

```
# logistic regression
model = LogisticRegression(penalty='l1', C=0.01, solver='liblinear', random_state=1)
```

## Summary

The most difficult part within this project is to deal with those data, not model itself. In the beginning, I didn't notice that the data is missing because what I used to open the csv file is VS code and it is hard to notice that there are some blanks within it. Data processing is important in this project. There is a lot of noise and missing values. Before finishing this project, I thought that parameters in the model or the model we chose are the key point during training, however, how we filter the data is more important. Besides, I search for several reference on Kaggle and combine others idea together to finish my homework. I think I learn a lot in this project.

## Comparisons of different approaches

Fill the missing values

method	private score
IterativeImputer	0.59082
IterativeImputer & LinearRegression for measurement_17	0.59128
SimpleImputer(strategy=most_frequent)	0.59103
SimpleImputer(strategy=most_frequent) & LinearRegression for measurement_17	<b>0.5916</b>
median	0.59097
mean	0.59082
KNNImputer(n_neighbors=5)	0.59048
KNNImputer(n_neighbors=15)	0.59087
KNNImputer(n_neighbors=20)	0.59075
KNNImputer(n_neighbors=15) & LinearRegression for measurement_17	0.59111

Model choosing

models	private score
LogisticRegression	<b>0.5916</b>
LinearRegression	0.5909

## Comprehensive related works survey

1. [Tabular Playground Series - Aug 2022 | Kaggle](#)  
The author mentions that attribute 2 and 3 looks like they are width and length dimension or similar.
2. [Tabular Playground Series - Aug 2022 | Kaggle](#)  
The author mentions that take measurements between 3 and 9, which have slightly higher correlation with measurement\_17, multiply each measurement by its correlation value with measurement\_17 and add it all up, we could get a new feature which is perfectly correlated with measurement\_17.
3. [TPSAUG22 EDA which makes sense ☆☆☆☆☆ | Kaggle](#)

The author mentions that our dataset is composed of many missing values so that we should do imputer to fill in those blank. Besides, we could observe the correlation among those columns through some chart.

## Thorough experimental results

Ablation study (SimpleImputer with most\_frequent, LogisticRegression)

standardization	check missing	drop measurement1	mean	private score
x	x	x	x	0.57968
v	x	x	x	0.59119
x	v	x	x	0.56921
x	x	v	x	0.57121
x	x	x	v	0.59013
v	v	x	x	0.5916
v	x	v	x	0.59114
v	x	x	v	0.59119
v	v	v	v	0.5916

LogisticRegression parameters

penalty	C	max_iter	solver	private score
l1	0.01	100	liblinear	0.5916
l2	0.01	100	lbfgs	0.59091
l2	0.01	100	sag	0.58353
l2	0.1	1000	sag	0.59092
l2	0.01	100	saga	0.56282
l2	0.1	1000	saga	0.5908

## Interesting findings or novel features engineering

There is a new column called “pred\_m\_17”, which represents prediction of measurement\_17 through Linear Regression. The purpose of this column is to record the values then I could fill the missing value more conveniently. Originally, I thought I

need to drop this column because there exists measurement\_17 without missing values, however, what surprise me is that the accuracy of not dropping this column is higher than the other one. I think it is because measurement\_17 is more important than the other column.

## Reference

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

<https://www.kaggle.com/code/anubhavde/tabular-playground-series-august2022>

<https://www.kaggle.com/competitions/tabular-playground-series-aug-2022/discussion/342319>

## Github Link

<https://github.com/james61124/james61124-Tabular-Playground-Series-Aug-2022-on-Kaggle>


## Environment details

<https://drive.google.com/file/d/1RYWyuHml7SKKxsV1oU7hkjb252ebncD2/view?usp=sharing>

## Weight Link

[https://drive.google.com/file/d/1OmG0MRq2z5xYJCMeAUWg0RI\\_ifmYuEyK/view?usp=share\\_link](https://drive.google.com/file/d/1OmG0MRq2z5xYJCMeAUWg0RI_ifmYuEyK/view?usp=share_link)

## Result

Overview Data Code Discussion Leaderboard Rules Team Submissions Late Submission			
<b>Submissions</b>			
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.			
Submissions evaluated for final score			
All Successful Selected Errors Recent			
Submission and Description	Private Score	Public Score	Selected
 submission (22).csv Complete (after deadline) · now	0.5916	0.58351	<input type="checkbox"/>