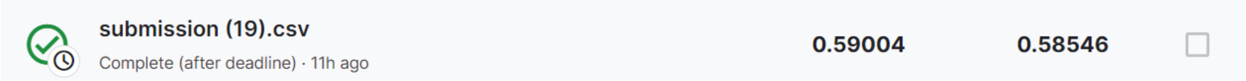
NCTU Introduction to Machine Learning, Final Project

109550134 梁詠晴

Introduction:

The task of the final project was to predict the product failure using the Tabular Playground Series from kaggle. The dataset contains much noise and missing values, therefore, I utilized several data preprocessing skills to improve the accuracy. Logistic regression model was used to predict the probability of product failure.

Screenshot of result:



submission (19).csv

Complete (after deadline) · 11h ago

0.59004 0.58546 □

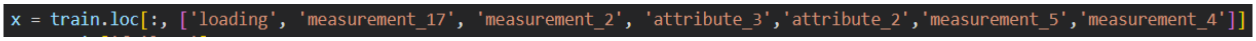
GitHub link:

https://github.com/sheepycat/NYCU_ML_final_project

Methodology:

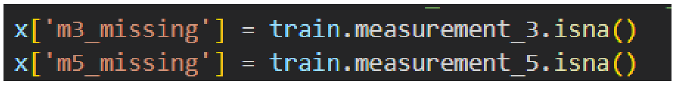
Data pre-process:

1. Inspired by TPSAUG22 EDA which makes sense, I noticed that the most of the features in the dataset are noise. Therefore, I tried different feature combinations and only preserve the features that would improve accuracy:



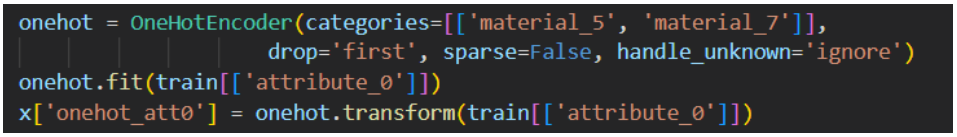
```
x = train.loc[:, ['loading', 'measurement_17', 'measurement_2', 'attribute_3','attribute_2','measurement_5','measurement_4']]
```

2. Inspired by Missing values have predictive value, I found that “whether the measurement value was missing” was related to the product failure. As a result, I add two features of missing values:



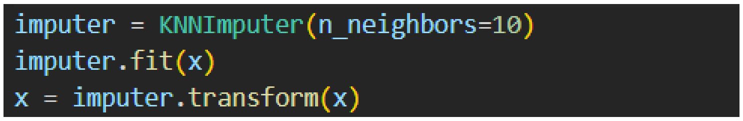
```
x['m3_missing'] = train.measurement_3.isna()
x['m5_missing'] = train.measurement_5.isna()
```

3. I used one-hot encoder to encode the categorical features as a one-hot numeric way:



```
onehot = OneHotEncoder(categories=[['material_5', 'material_7']],
                       drop='first', sparse=False, handle_unknown='ignore')
onehot.fit(train[['attribute_0']])
x['onehot_att0'] = onehot.transform(train[['attribute_0']])
```

4. I used KNNImputer to complete missing values in the data using k-Nearest Neighbors.

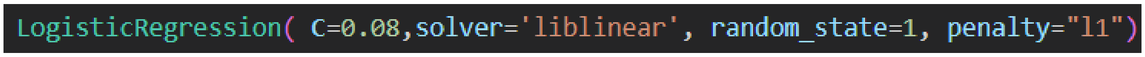


```
imputer = KNNImputer(n_neighbors=10)
imputer.fit(x)
x = imputer.transform(x)
```

5. Inspired by Less can be more: Feature Engineering Ideas, I multiply attribute 2 and 3 to get the area.
6. Used StandardScaler() for dataset Standardization

Model architecture and Hyperparameters:

- LogisticRegression()
  - C = 0.08
  - Solver = ‘liblinear’
  - Random_state = 1
  - Penalty = “l1”



```
LogisticRegression( C=0.08,solver='liblinear', random_state=1, penalty="l1")
```

Reference:

1. TPSAUG22 EDA which makes sense
2. Missing values have predictive value
3. Less can be more: Feature Engineering Ideas

Summary:

By using data preprocessing skills and logistic regression model, I was able to score 0.59004 on the private dataset. Data pre-process is important in this project, training on useful data only improved accuracy a lot. In real world data, it is also important to understand the meaning of the features, so that we can aggregate relevant features and generate meaningful features.