

# Machine Learning Operation(MLOps)

- Ch2. Level0 MLOps(1)

## Level0 MLOps(1)

---

1. Train / Inference script
2. SQLite3
3. Dash

# 데이터셋

The screenshot shows the Kaggle dataset page for 'Steel Plates Faults Detection' by Bhushan Kapkar, updated 3 years ago. The page features a sidebar with navigation options like Home, Competitions, Datasets, Models, Code, Discussions, Learn, and More. The main content area includes a search bar, a 'New Notebook' button, and a 'Download (105 kB)' button. The dataset title 'Steel Plates Faults Detection' is prominently displayed, followed by a green and yellow patterned image. Below the title, there are tabs for 'Data Card', 'Code (0)', 'Discussion (0)', and 'Suggestions (0)'. The 'About Dataset' section describes the task of predicting defect detection of steel plates and provides a link to a research paper. The 'Usability' is 4.12, and the 'License' is Unknown. The 'Expected update frequency' is Not specified. The 'Tags' include Business, Computer Science, Law, India, and Manufacturing.

Steel Plates Faults Detection

12 New Notebook Download (105 kB)

Data Card Code (0) Discussion (0) Suggestions (0)

### About Dataset

Propose a Machine Learning Model to predict/detect the defect detection of steel plates from the dataset (attached in the annexure).

Use the Variable descriptor file to study about the variables used in the dataset.

The proposed Model shall be justified by comparison analysis with different algorithms such as Random Forest, SVM, K Nearest Neighbour, Logistic Regression, Decision Trees, Gaussian Naïve Bayes with suitable graphs/charts.

Acknowledgements

MetaNet: The Theory of Independent Judges (PDF Download Available). Available from: [https://www.researchgate.net/publication/13731626\\_MetaNet\\_The\\_Theory\\_of\\_Independent\\_Judges](https://www.researchgate.net/publication/13731626_MetaNet_The_Theory_of_Independent_Judges) [accessed Sep 6, 2017].

Dataset provided by Semeion, Research Center of Sciences of Communication, Via Sersale 117, 00128, Rome, Italy.

[www.semeion.it](http://www.semeion.it)

Lichman, M. (2013). UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science.

Usability 4.12

License Unknown

Expected update frequency Not specified

Tags: Business, Computer Science, Law, India, Manufacturing

<https://www.kaggle.com/datasets/bpkapkar/steel-plates-faults-detection?select=Steel+Plates+Faults+Data+Set.csv>

## 학습/검증 – 1.analysis.ipynb

```
import pandas as pd
import numpy as np
```

```
## data load
```

```
dat = pd.read_csv("train.csv")
```

← 학습용파일 로드

```
## train, test split
```

```
from sklearn.model_selection import train_test_split
```

```
train, test = train_test_split(dat, test_size=0.3)
```

```
train.head()
```

← Train/Test 분할

```
## x variables preprocessing
```

```
x_cols = ['V'+str(i) for i in range(1,28)]
```

```
from sklearn.preprocessing import StandardScaler
```

```
trans = StandardScaler()
```

```
trans.fit(train[x_cols])
```

```
train_x = trans.transform(train[x_cols])
```

← X변수 표준화

```
## y variables preprocessing
```

```
train['V34'] = train['Class']-1
```

```
train_y = [str(np.where(r==1)[0][0]) for r in train[['V'+str(i) for i in range(28,35)]].to_numpy()]
```

← y변수 변환

## 학습/검증 – 1.analysis.ipynb

```
## classification modeling
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
model.fit(train_x, train_y)
```

← 분류 모델링(RandomForest)

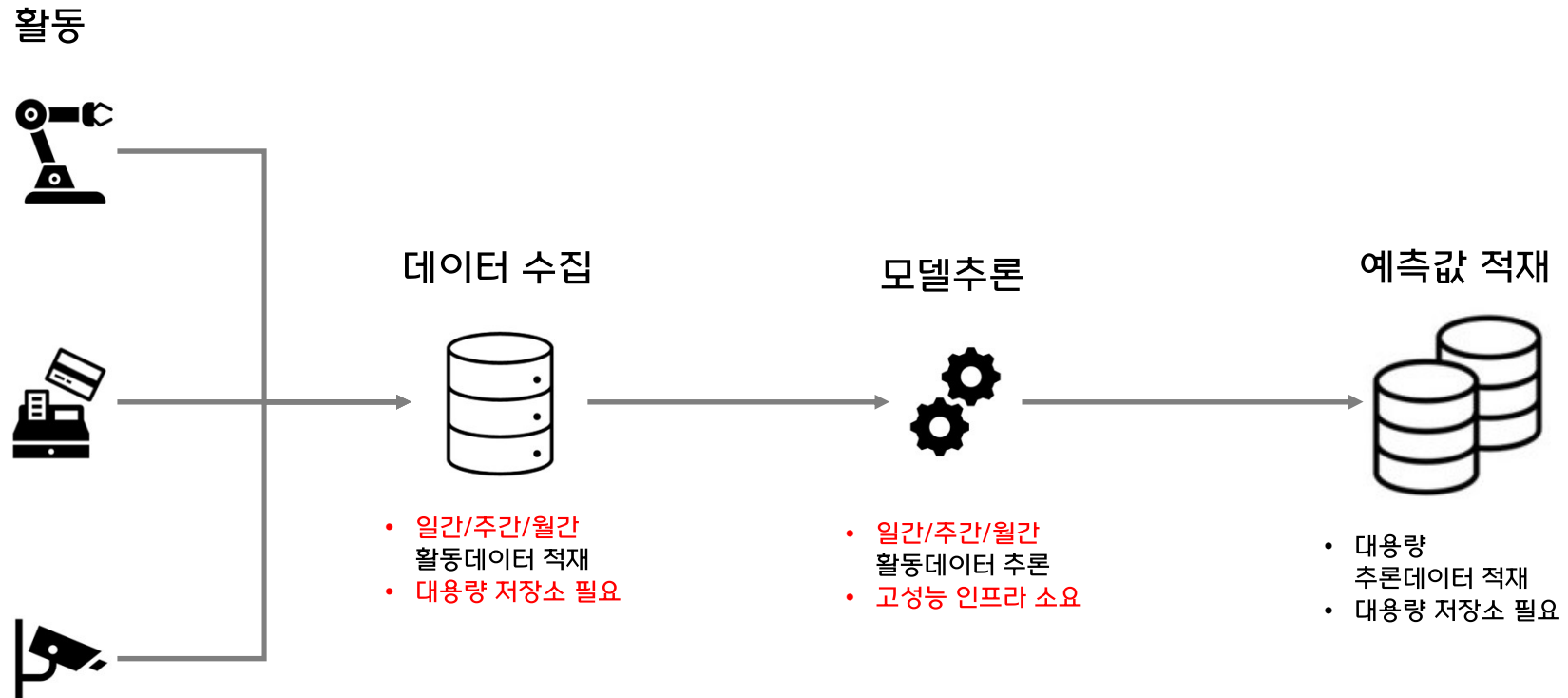
```
## make prediction for testset
test_x = trans.transform(test[x_cols])
pred = model.predict(test_x)
```

← Test셋 예측값 도출

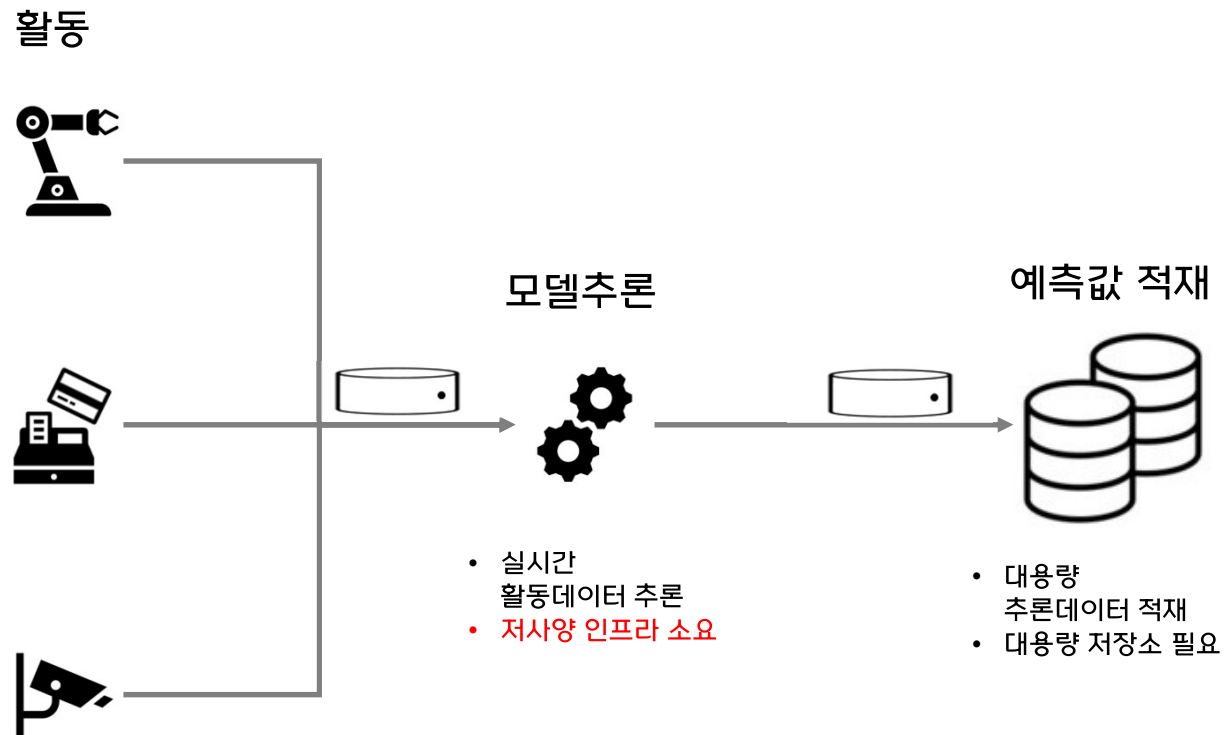
```
## validation for testset
test['V34'] = test['Class']-1
test_y = [str(np.where(r==1)[0][0]) for r in test[['V'+str(i) for i in range(28,35)]].to_numpy()]
np.mean(pred==test_y)
pd.crosstab(test_y, pred)
```

← Test셋 정확도 검증

## 배치 추론과 실시간 추론 (Online and Batch serving)



## 배치 추론과 실시간 추론 (Online and Batch serving)



## 배치 추론 – batch\_inference.py

```
import pandas as pd
import numpy as np
from sklearn.ensemble import
RandomForestClassifier
import pickle as pkl

## load preprocessing and model
trans = pkl.load(open("trans.pkl","rb"))
model = pkl.load(open("model.pkl","rb"))

## load new dataset
new_data = pd.read_csv("test.csv")
x_cols = ['V'+str(i) for i in range(1,28)]

## prediction for new dataset
x = trans.transform(new_data[x_cols])
pred = model.predict(x)
```

← 전처리 및 모델 로딩

← 전처리 적용

← 예측값 산출



## 실시간 추론 – online\_inference.py

```
import pandas as pd
import numpy as np
import pickle as pkl
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
```

```
## load preprocessing and model
trans = pkl.load(open("trans.pkl","rb"))
model = pkl.load(open("model.pkl","rb"))
```



전처리 및 모델 로딩

```
## define function to get new data
def select_row(id):
    a=id-1
    dat = pd.read_csv("test.csv")
    row = dat[a:id]
    return row
```



데이터 추출 함수 정의

```
## preprocessing for new data
new_data = select_row(1)
x_cols = ['V'+str(i) for i in range(1,28)]
x = trans.transform(new_data[x_cols])
```



전처리 적용

```
## prediction for new data
pred = model.predict(x)
```



예측값 산출

End of Document