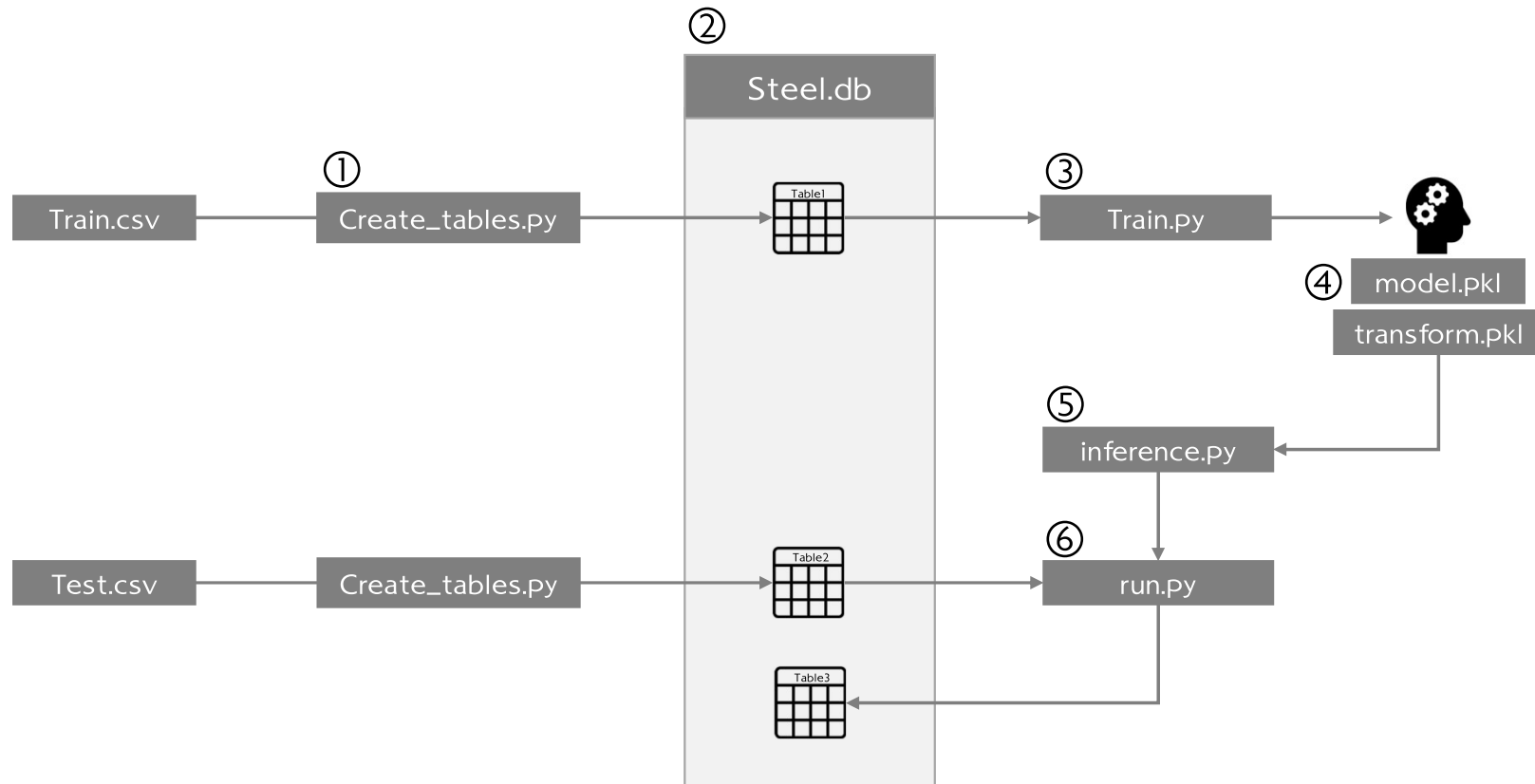


Machine Learning Operation(MLOps)

- Ch2. Level0 MLOps(3), SQLite3

학습/추론 코드 수정

기본 흐름도 정의



db생성 – create_tables.py

```
import sqlite3
import pandas as pd

db_name = "steel.db"
conn = sqlite3.connect(db_name)
c = conn.cursor()

### create train table into DB ###
train = pd.read_csv("train.csv")
train.to_sql("train", conn, index=False)

### create test table into DB ###
test = pd.read_csv("test.csv")
test.to_sql("test", conn, index=False)

### create predict table ###
c.execute("""CREATE TABLE predict (
                predict REAL
            )""")
conn.commit()
```

```
print("\n\n train table...")
c.execute("SELECT * FROM train")
items = c.fetchall()
for item in items:
    print(item)
```

```
print("\n\n test table...")
c.execute("SELECT * FROM test")
items = c.fetchall()
for item in items:
    print(item)
```

```
print("\n\n predict table...")
c.execute("SELECT * FROM predict")
items = c.fetchall()
for item in items:
    print(item)
```

```
## close our connection
conn.close()
```

모델 생성 – train.py

```
import sqlite3
import pandas as pd
import numpy as np
import pickle as pkl

db_name = "steel.db"
conn = sqlite3.connect(db_name)
c = conn.cursor()
c.execute("SELECT * FROM train")
cols = [col[0] for col in c.description]
dat = pd.DataFrame(data=c.fetchall(), columns=cols)
conn.close()

## train, test split
from sklearn.model_selection import train_test_split
train, test = train_test_split(dat, test_size=0.3)
train.head()

## 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])

## 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())
```

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

## save model and transformation
print("creating model file...")
pkl.dump(model, open("model.pkl", "wb"))
print("creating transformation file...")
pkl.dump(trans, open("transform.pkl", "wb"))

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

## 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()]
print(np.mean(pred==test_y))
print(pd.crosstab(test_y, pred))
```

추론함수 정의 – inference.py

```
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
import pickle as pkl

def inference(dat):
    model = pkl.load(open("model.pkl","rb"))
    trans = pkl.load(open("transform.pkl","rb"))

    ## x variables preprocessing
    x_cols = ['V'+str(i) for i in range(1,28)]
    test_x = trans.transform(dat[x_cols])

    ## make prediction for testset
    pred = model.predict(test_x)
    return pred
```

흐름 실행(배치추론) – batch_run.py

```
import sqlite3
import pandas as pd
import numpy as np
import inference as inf

## load test dataset
print("\n\nLoad test table...")

db_name = "steel.db"
conn = sqlite3.connect(db_name)
c = conn.cursor()
c.execute("SELECT * FROM test")
cols = [col[0] for col in c.description]
dat = pd.DataFrame(data=c.fetchall(), columns=cols)
conn.close()

## make prediction for test set
print("\n\nmake prediction...")

pred = inf.inference(dat)
```

```
## insert prediction into predict table
print("\n\ninsert predict table...")

db_name = "steel.db"
conn = sqlite3.connect(db_name)
c = conn.cursor()
c.executemany("INSERT INTO predict VALUES (?)", pred)
conn.commit()

## print predict table
print("\n\npredict table...")

c.execute("SELECT * FROM predict limit 10")
items = c.fetchall()
for item in items:
    print(item)

conn.close()
```

흐름 실행(실시간 추론) – online_run.py

```
import sqlite3
import pandas as pd
import numpy as np
import inference as inf
import sys

## Inputing argument
rid = sys.argv[1]
print("{}-th row need to be predicted".format(rid))

## extract one-row of data
db_name = "steel.db"
conn = sqlite3.connect(db_name)
c = conn.cursor()
c.execute("SELECT * FROM test where rowid = {}".format(rid))
cols = [col[0] for col in c.description]
dat = pd.DataFrame(data=c.fetchall(), columns=cols)
conn.close()

## prediction
pred = inf.inference(dat)

print(pred)
```


End of Document