Feast_NNI_BentoML_DB

210901

AI_Ops

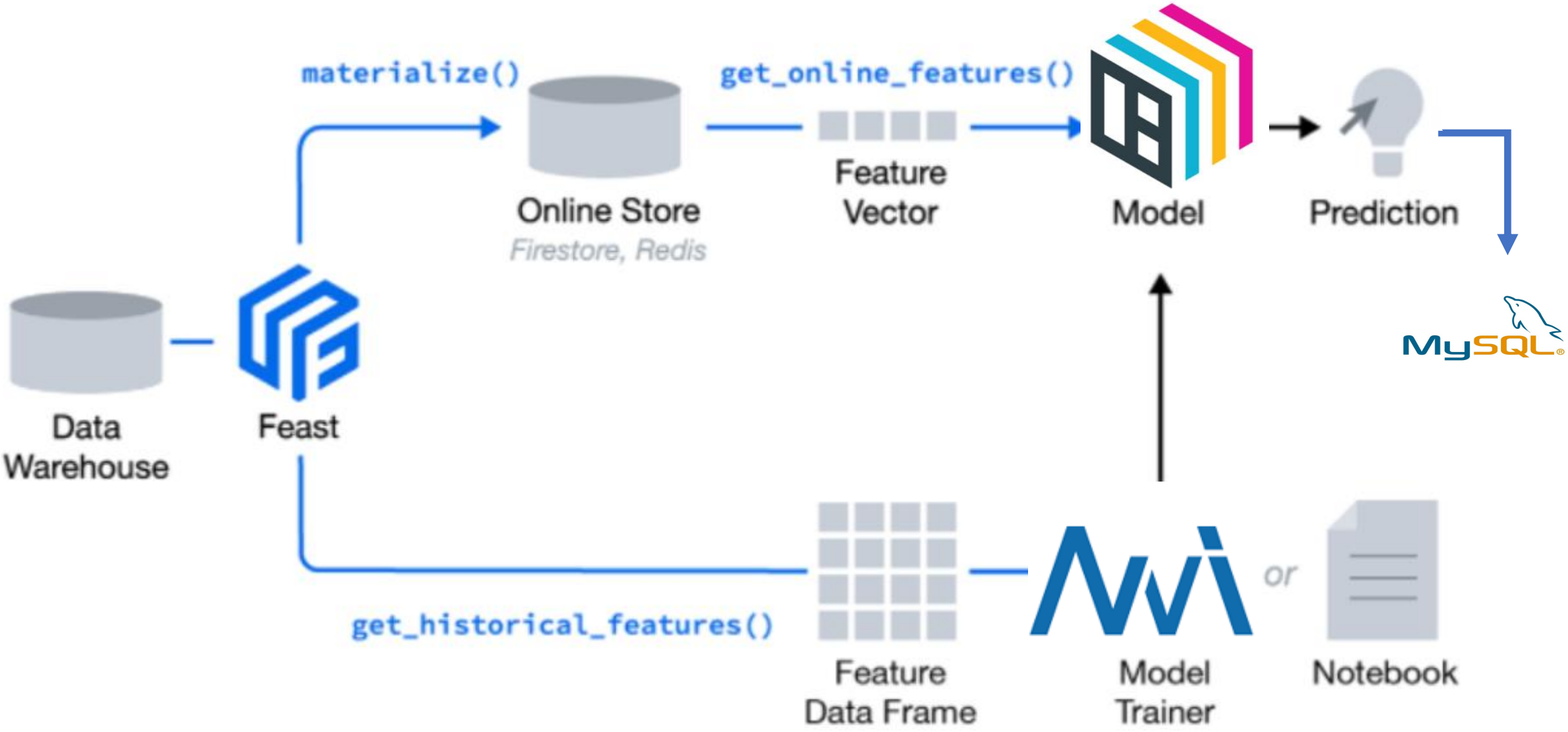
이민준

FEAST

Feast_


materialize()
get_online_features()
Online Store
Firestore, Redis
Feature
Vector
Model
Prediction
Data
Warehouse
Feast
get_historical_features()
Feature
Data Frame
Model
Trainer
or
Notebook

Feast_


materialize()
get_online_features()
Online Store
Firestore, Redis
Feature
Vector
Model
Prediction
Data
Warehouse
Feast
MySQL
get_historical_features()
Feature
Data Frame
Model
Trainer
or
Notebook

Feast_

Kafka_spark


kafka
벌써 자면 어떡해
kafka
뭐하는 거야..
kafka
일어나
APACHE Spark
24
전처리 해야지


APACHE Spark
전처리 중
APACHE Spark
전처리 중
역시 형이야 구하러 왔구나
APACHE Spark
아니 나도 전처리하러 왔어

Feast_

Feast_

Feast_

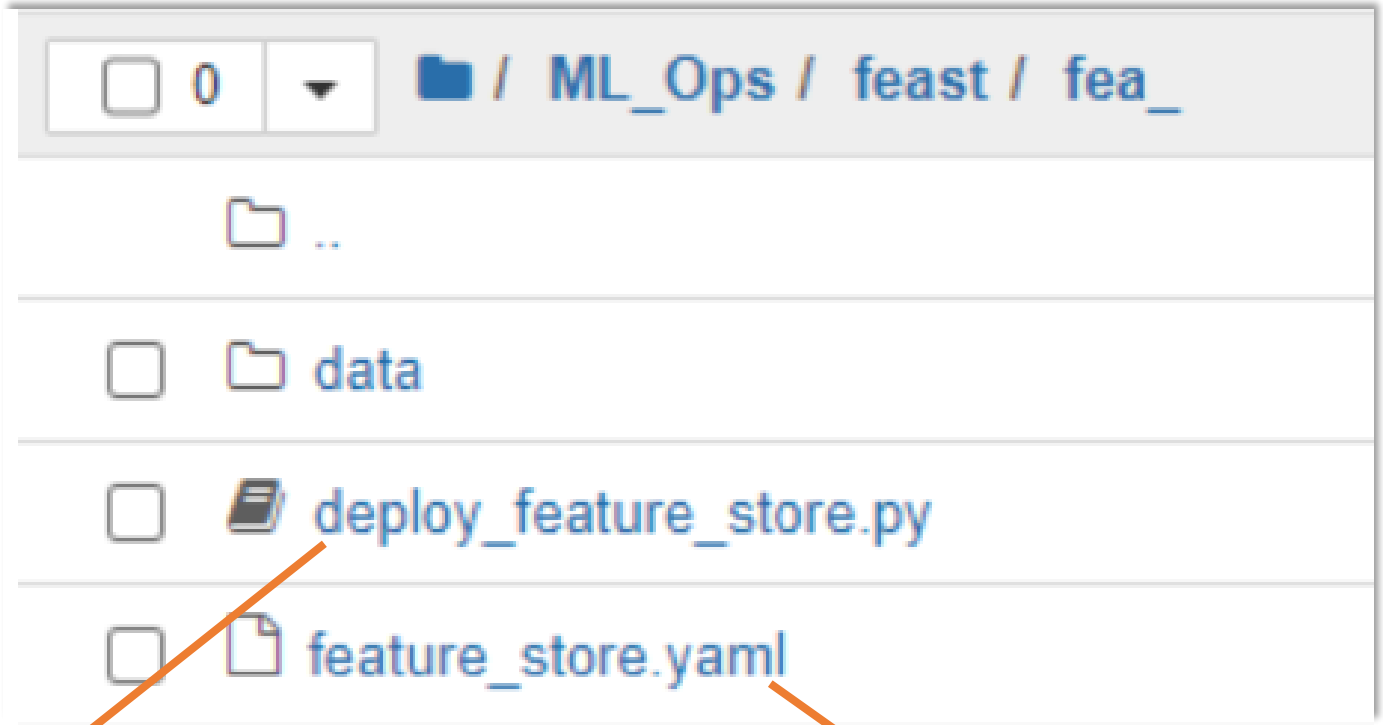
ppr_data_.parquet

아직은 parquet만 지원

Feast_

```
> feast init fea_
```

- 'fea_'

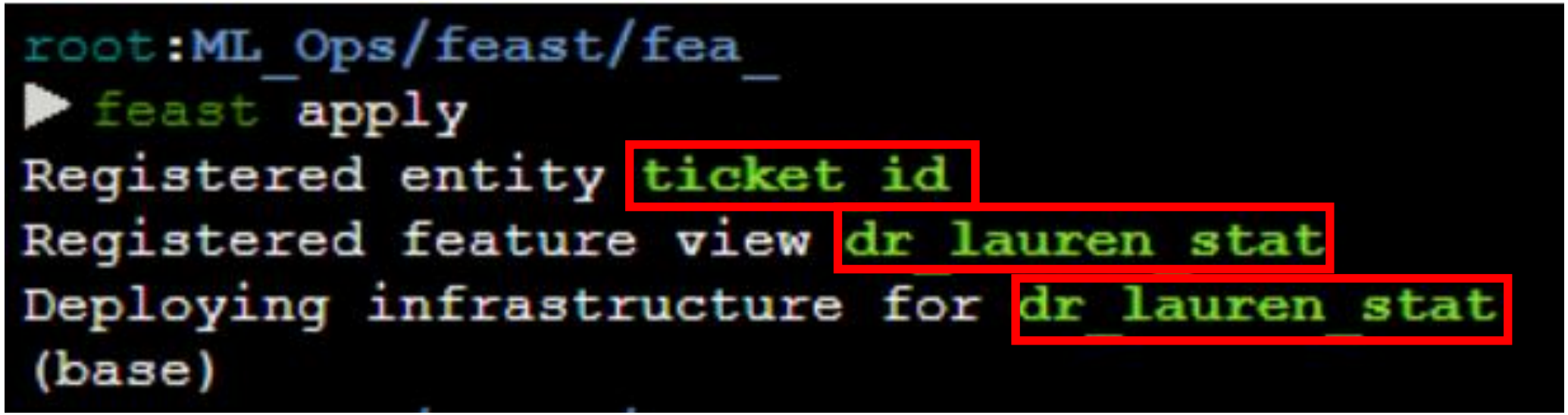



```
dr_lauren_stat = FileSource(
    path="/workspace/ML_Ops/feast/fea_/data/ppr_data_.parquet",
    event_timestamp_column="event_timestamp",
```

```
dr_lauren_view = FeatureView(
Feature(name= columns_name, dtype = ~), x num_cols
)
```

```
project: fea_
registry: data/registry.db
provider: local
online_store:
    path: data/online_store.db
```

Feast_


root:ML_Ops/feast/fea_
feast apply
Registered entity ticket id
Registered feature view dr lauren stat
Deploying infrastructure for dr_lauren_stat
(base)

Feast_

Feast_nni


Data
Warehouse
Feast
get_historical_features()
Feature
Data Frame
Model
Trainer

Feast_nni

```
### entity_df
parquet_ = pd.read_parquet(f'{path_}/data/ppr_data_.parquet', engine='pyarrow')
orders = parquet_[['ticket_id','event_timestamp']]

# Retrieve training data
training_df = fs.get_historical_features(
    entity_df=orders,
    features=[
        "dr_lauren_stat:time",
        "dr_lauren_stat:weekday",
        "dr_lauren_stat:weekend",
        "dr_lauren_stat:instlo_1",
        "dr_lauren_stat:instlo_2",
        "dr_lauren_stat:inst_code",
        "dr_lauren_stat:sysname_lo",
        "dr_lauren_stat:sysname_eq",
        "dr_lauren_stat:ntt_label",
    ],
).to_df()
```

Feast_nni

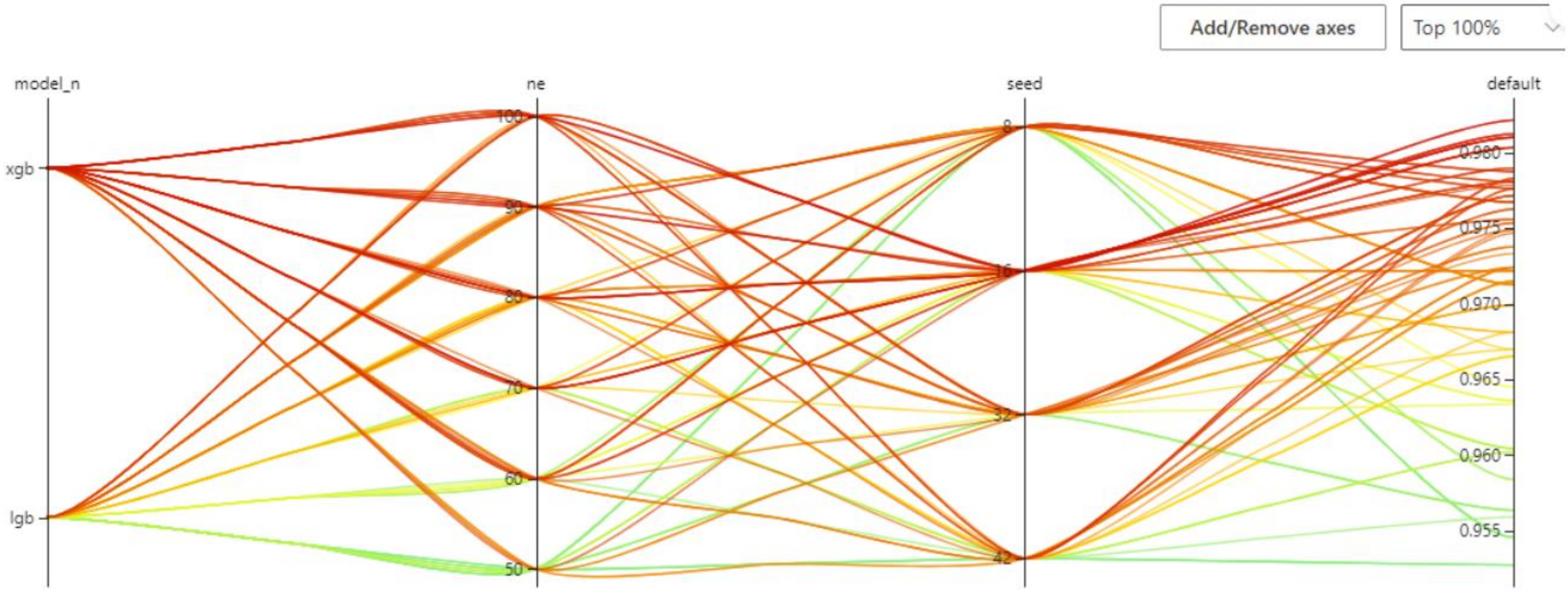
	event_timestamp	ticket_id	time	weekday	weekend	instlo_1	instlo_2	inst_code	sysname_lo	sysname_eq	ntt_label
0	2021-01-01 03:17:00+00:00	3604002	3.283333	4	0	9	47	531	117	0	1
1	2021-01-01 03:17:00+00:00	3604002	3.283333	4	0	9	47	531	117	0	1
2	2021-01-01 03:20:00+00:00	3604035	3.333333	4	0	12	84	340	913	0	1
3	2021-01-01 03:20:00+00:00	3604035	3.333333	4	0	12	84	340	913	0	1
4	2021-01-02 08:38:00+00:00	3624904	8.633333	5	1	10	101	435	868	0	1
...	...	...	...	...	...	...	...	...	...	...	...
14730	2021-06-30 18:33:00+00:00	9346264	18.550000	2	0	2	67	400	173	0	0
14731	2021-06-30 18:35:00+00:00	9346310	18.583333	2	0	2	67	375	1506	0	0
14732	2021-06-30 18:35:00+00:00	9346329	18.583333	2	0	2	67	382	1978	0	0
14733	2021-06-30 18:40:00+00:00	9346522	18.666667	2	0	2	122	1504	161	0	0
14734	2021-06-30 18:40:00+00:00	9346522	18.666667	2	0	2	122	1504	161	0	0

Neural Network Intelligence

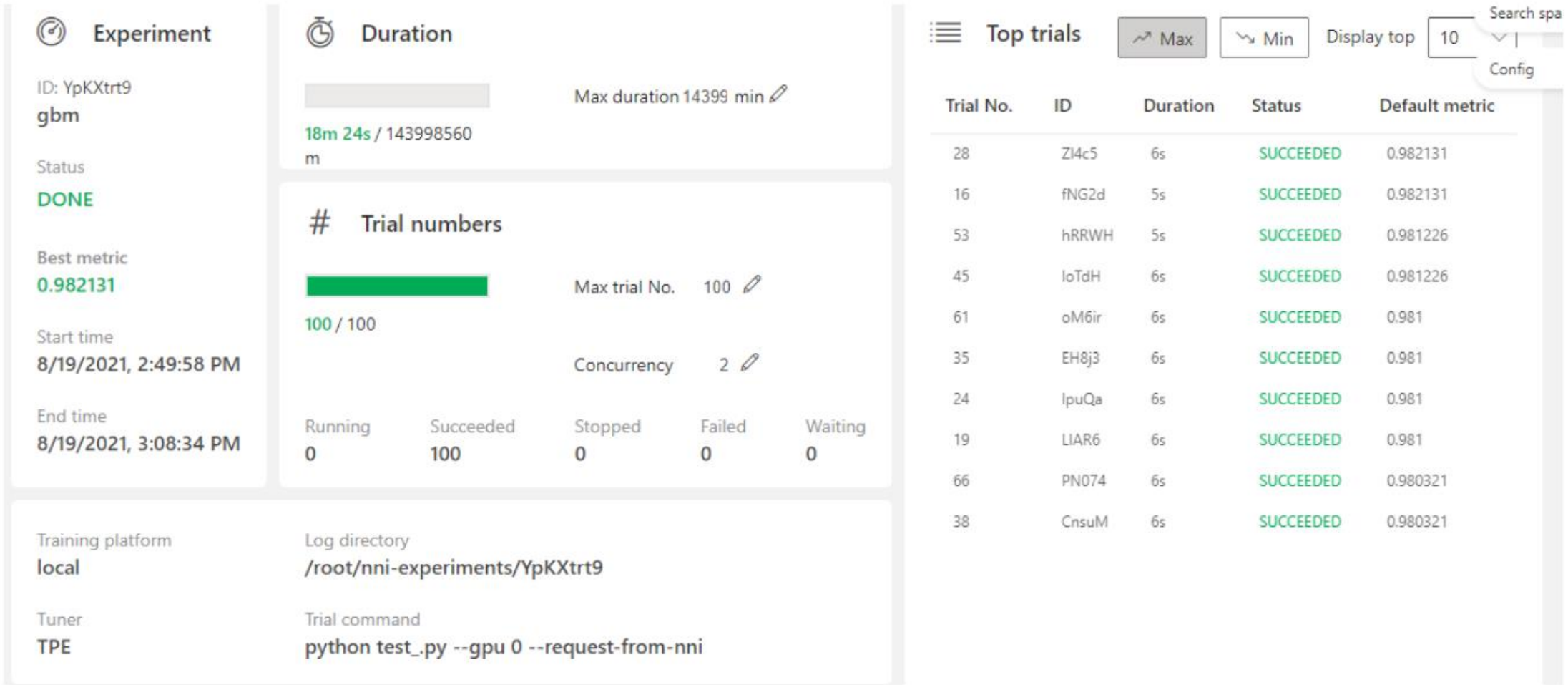
nni


```
{
  "model_n": {
    "_type": "choice",
    "_value": ["lgb", "xgb"]
  },
  "ne": {
    "_type": "choice",
    "_value": [50, 60, 70, 80, 90, 100]
  },
  "seed": {
    "_type": "choice",
    "_value": [42, 32, 16, 8]
  }
}
```

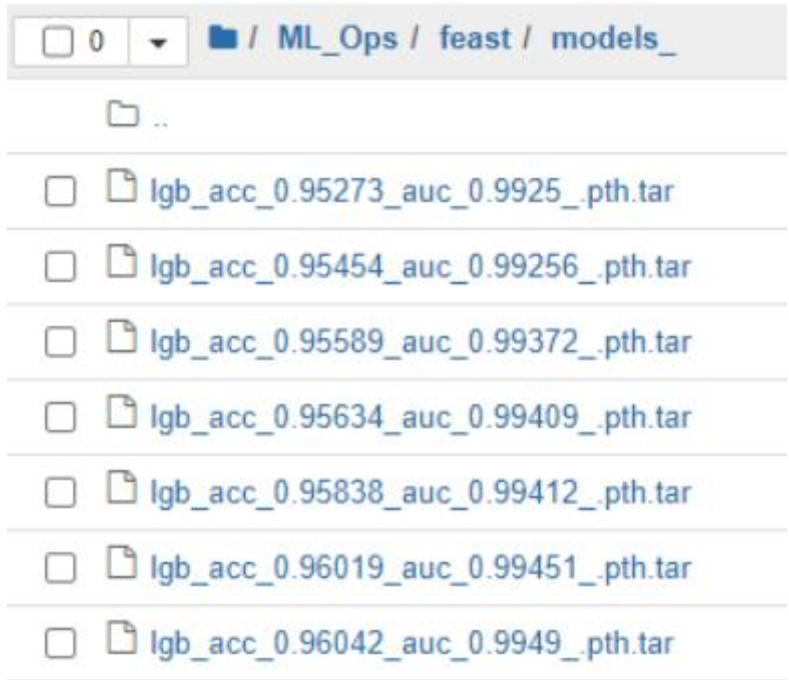
nni


Add/Remove axes
Top 100%
model_n
ne
seed
default
xgb
lgb
100
90
80
70
60
50
8
16
32
42
0.980
0.975
0.970
0.965
0.960
0.955

nni

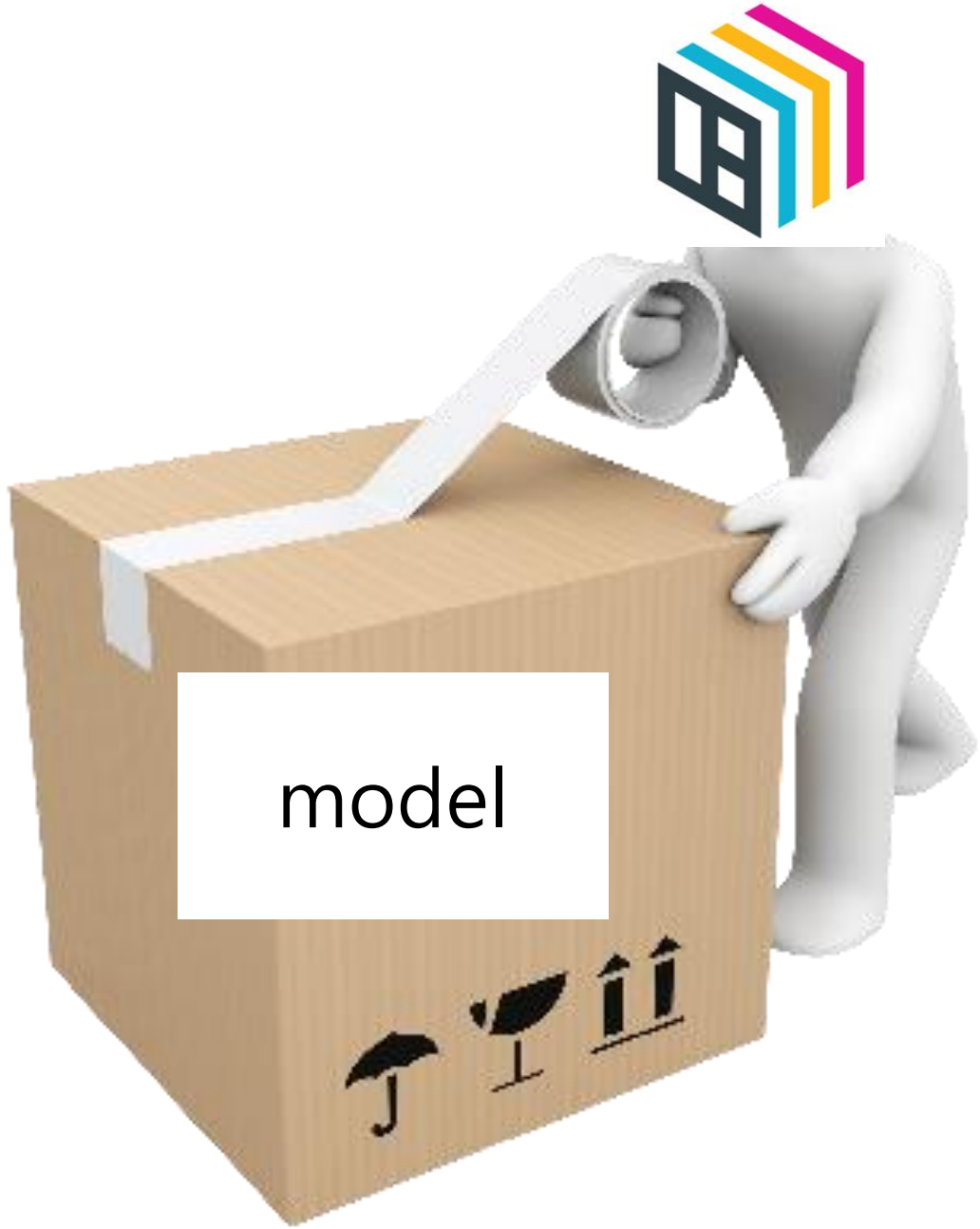

Experiment
ID: YpKXtrt9
gbm
Status
DONE
Best metric
0.982131
Start time
8/19/2021, 2:49:58 PM
End time
8/19/2021, 3:08:34 PM
Duration
Max duration 14399 min
18m 24s / 143998560
m
Trial numbers
Max trial No. 100
100 / 100
Concurrency 2
Running
0
Succeeded
100
Stopped
0
Failed
0
Waiting
0
Training platform
local
Log directory
/root/nni-experiments/YpKXtrt9
Tuner
TPE
Trial command
python test_.py --gpu 0 --request-from-nni
Top trials
Max
Min
Display top 10
Search spa
Config
Trial No. ID Duration Status Default metric
28 ZI4c5 6s SUCCEEDED 0.982131
16 fNG2d 5s SUCCEEDED 0.982131
53 hRRWH 5s SUCCEEDED 0.981226
45 IoTdH 6s SUCCEEDED 0.981226
61 oM6ir 6s SUCCEEDED 0.981
35 EH8j3 6s SUCCEEDED 0.981
24 IpuQa 6s SUCCEEDED 0.981
19 LIAR6 6s SUCCEEDED 0.981
66 PN074 6s SUCCEEDED 0.980321
38 CnsuM 6s SUCCEEDED 0.980321

nni


0
/ ML_Ops / feast / models_
..
lgb_acc_0.95273_auc_0.9925_.pth.tar
lgb_acc_0.95454_auc_0.99256_.pth.tar
lgb_acc_0.95589_auc_0.99372_.pth.tar
lgb_acc_0.95634_auc_0.99409_.pth.tar
lgb_acc_0.95838_auc_0.99412_.pth.tar
lgb_acc_0.96019_auc_0.99451_.pth.tar
lgb_acc_0.96042_auc_0.9949_.pth.tar

BENTOML

BentoML



BentoML

Pack.py

```
@env(infer_pip_packages=True)
@artifacts([SklearnModelArtifact('model')])
class Dr_lauren_classifier(BentoService):
    """
    A minimum prediction service exposing a Scikit-learn model
    """

    @api(input=DataframeInput(), batch=True)
    def predict(self, df: pd.DataFrame):
        """
        An inference API named `predict` with Dataframe input adapter, which codifies
        how HTTP requests or CSV files are converted to a pandas Dataframe object as the
        inference API function input
        """
        return self.artifacts.model.predict(df)
```

Package_save.py

```
# create dr_lauren_service instance
dr_lauren_service = pack_.Dr_lauren_classifier()

# get_model
mo_ = torch.load('/workspace/ML_Ops/feast/fea_/models_/xgb_acc_0.97829_auc_0.99705_.pth.tar')['model']

# dr_lauren_classifier와 'model'로 패키징됨
dr_lauren_service.pack('model', mo_)

# 경로 저장
saved_path = dr_lauren_service.save()
```

BentoML

```
> Python Package_save.py
```

BentoML

```
root:ML_Ops/feast/fea_
▶ ls /root/bentoml/repository/Dr_lauren_classifier
20210819063311_8F9A48  20210819074959_34A218  20210819075428_1FF3A7  20210819080455_1C8532
(base)
root:ML_Ops/feast/fea_
▶ ls /root/bentoml/repository/Dr_lauren_classifier/20210819080455_1C8532
bentoml-init.sh  docker-entrypoint.sh  docs.json            environment.yml  python_version  requirements.txt
bentoml.yml      Dockerfile            Dr_lauren_classifier  MANIFEST.in      README.md       setup.py
(base)
```

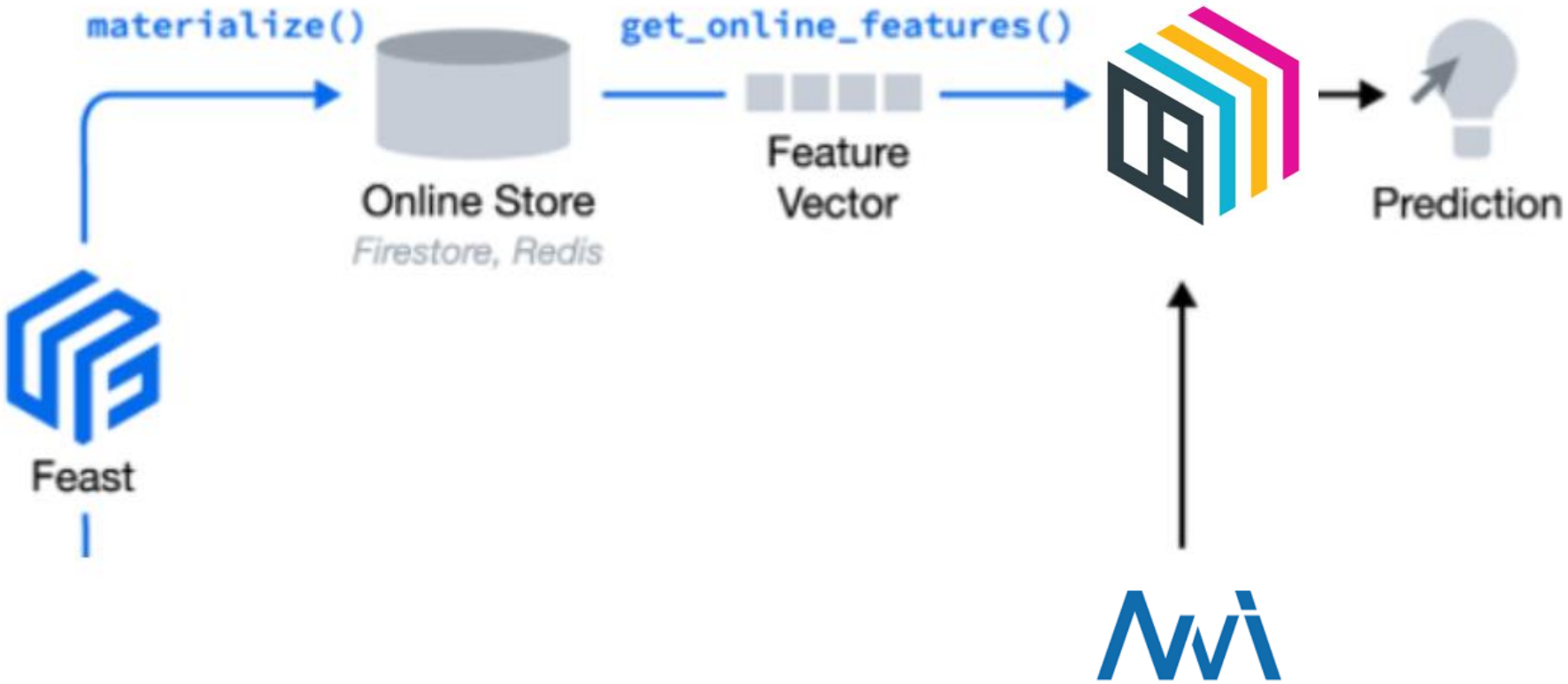
BentoML

```
> Bentoml serving Dr_lauren_classifier:latest
```

BentoML

```
root:/workspace/ML_Ops/feast
▶ bentoml serve Dr_lauren_classifier:latest
[2021-08-22 08:48:12,763] INFO - Getting latest version Dr_lauren_classifier:20210819080455_1C8532
[2021-08-22 08:48:12,775] INFO - Starting BentoML API proxy in development mode..
[2021-08-22 08:48:12,778] INFO - Starting BentoML API server in development mode..
[2021-08-22 08:48:12,975] INFO - Micro batch enabled for API `predict` max-latency: 20000 max-batch-size 4000
[2021-08-22 08:48:12,975] INFO - Your system nofile limit is 1048576, which means each instance of microbatch service is ab
le to hold this number of connections at same time. You can increase the number of file descriptors for the server process,
 or launch more microbatch instances to accept more concurrent connection.
======== Running on http://0.0.0.0:5000 ========
(Press CTRL+C to quit)
 * Serving Flask app "Dr_lauren_classifier" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:51222/ (Press CTRL+C to quit)
```

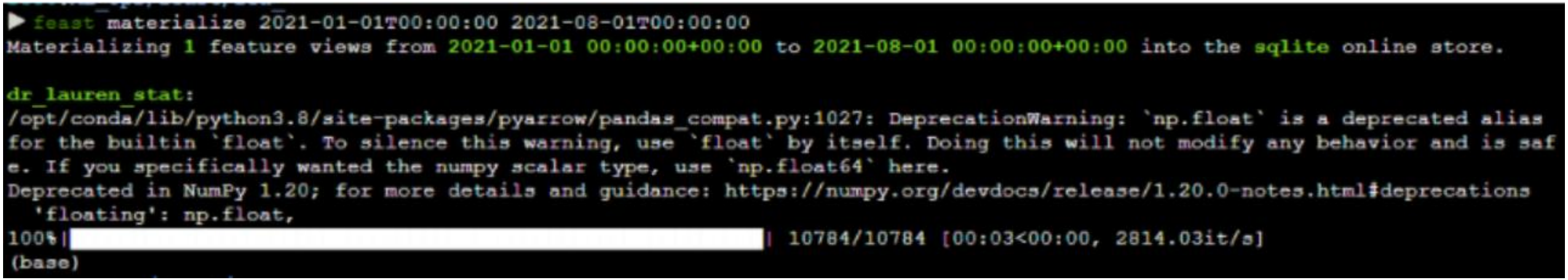
Feast_bentoml



Feast_bentoml

```
> Feast materialize 2021-01-01T00:00:00 2021-08-01T00:00:00
```

Feast_bentoml



ticket_id	event_timestamp
9360646	2021-07-01 02:47:00
9361358	2021-07-01 03:11:00
9363518	2021-07-01 04:25:00
9365171	2021-07-01 05:29:00
9365176	2021-07-01 05:29:00
...	...
9995149	2021-07-15 18:37:00
9995205	2021-07-15 18:39:00
9995206	2021-07-15 18:39:00
9995241	2021-07-15 18:40:00
9995586	2021-07-15 18:49:00

```
def get_entities(path_,from_time_):
    parquet_ = pd.read_parquet(path_, engine='pyarrow')
    orders = parquet_[['ticket_id','event_timestamp']]

    ### 뽑을 rows의 entity == key 라고 볼 수 있음
    new_orders = orders[orders['event_timestamp']>= from_time_]
    new_orders.drop_duplicates(ignore_index=True, inplace=True)
    return new_orders
```

Test_entity

Feast_bentoml

```
online_ = fs_.get_online_features(
            entity_rows=[{"ticket_id": i} for i in entity_df_['ticket_id']],
            features=[
                "dr_lauren_stat:time",
                "dr_lauren_stat:weekday",
                "dr_lauren_stat:weekend",
                "dr_lauren_stat:instlo_1",
                "dr_lauren_stat:instlo_2",
                "dr_lauren_stat:inst_code",
                "dr_lauren_stat:sysname_lo",
                "dr_lauren_stat:sysname_eq",
                "dr_lauren_stat:ntt_label",
        ],
)
```

Feast_bentoml

	time	weekday	weekend	instlo_1	instlo_2	inst_code	sysname_lo	sysname_eq	ntt_label
0	2.783333	3	0	3	15	781	1584	0	1
1	3.183333	3	0	12	38	693	972	0	1
2	4.416667	3	0	3	53	293	2200	0	1
3	5.483333	3	0	11	37	91	834	0	1
4	5.483333	3	0	11	37	525	820	0	1
...	...	...	...	...	...	...	...	...	...
1368	18.616667	3	0	2	67	400	173	0	0
1369	18.650000	3	0	2	67	375	1506	0	0
1370	18.650000	3	0	2	67	382	1978	0	0
1371	18.666667	3	0	9	65	642	2061	0	1
1372	18.816667	3	0	14	128	1215	2047	0	1

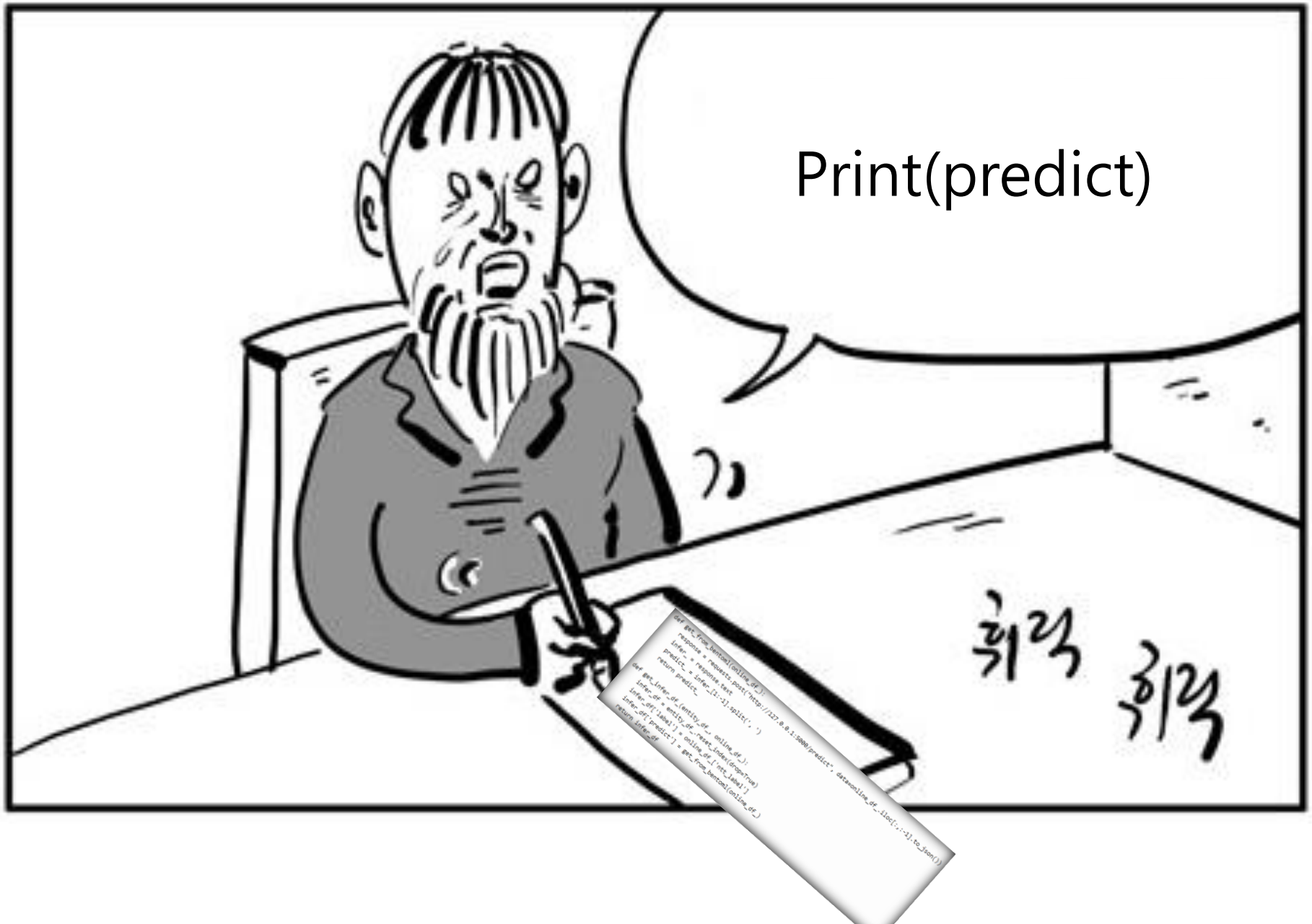
Request

Feast_bentoml

```
def get_from_bentoml(online_df_):
    response = requests.post("http://127.0.0.1:5000/predict", data=online_df_.iloc[:,:-1].to_json())
    infer_ = response.text
    predict_ = infer_[1:-1].split(', ')
    return predict_

def get_infer_df_(entity_df_, online_df_):
    infer_df = entity_df_.reset_index(drop=True)
    infer_df['label'] = online_df_['ntt_label']
    infer_df['predict'] = get_from_bentoml(online_df_)
    return infer_df
```

Feast_bentoml



Feast_bentoml

Feast_bentoml

```
▶ python online_inf.py
/opt/conda/lib/python3.8/site-packages/pyarrow/pandas_compat.py:1027: DeprecationWarning: `np.float` is a deprecated alias
for the builtin `float`. To silence this warning, use `float` by itself. Doing this will not modify any behavior and is saf
e. If you specifically wanted the numpy scalar type, use `np.float64` here.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
  'floating': np.float,
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
```

To_DB


get_online_features()
Feature
Vector
Model
Prediction
MySQL

Feast_bentoml_DB

```
### data create, insert, update
def get_commit_sql(conn_, sql_):
    with conn_.cursor() as cursor:
        cursor.execute(sql_)
        result = cursor.fetchall()
        conn.commit()

### data read
def get_data_sql(conn_, sql_):
    with conn_.cursor() as cursor:
        cursor.execute(sql_)
        result = cursor.fetchall()
        return result
```

```
def get_columns_(df_):
    columns_str = ', '.join([i for i in infer_df.columns])
    return columns_str


def get_values_(df_):
    value_ = []
    for row_ in df_.values:
        value_.append("("+", ".join(["'"+str(i)+"'" for i in row_]) + ")")

    values_str = ', '.join(value_)
    return values_str
```

Feast_bentoml_DB

Feast_bentoml_DB

Feast_bentoml
Feast_bentoml

```
mysql> select * from infer_df limit 30;
+-----------+---------------------+-------+---------+
| ticket_id | event_timestamp     | label | predict |
+-----------+---------------------+-------+---------+
|   9360646 | 2021-07-01 02:47:00 |     1 |       1 |
|   9361358 | 2021-07-01 03:11:00 |     1 |       1 |
|   9363518 | 2021-07-01 04:25:00 |     1 |       1 |
|   9365171 | 2021-07-01 05:29:00 |     1 |       1 |
|   9365176 | 2021-07-01 05:29:00 |     1 |       1 |
|   9368107 | 2021-07-01 07:27:00 |     1 |       1 |
|   9370665 | 2021-07-01 08:58:00 |     1 |       1 |
|   9370880 | 2021-07-01 09:07:00 |     1 |       1 |
|   9370921 | 2021-07-01 09:07:00 |     1 |       1 |
|   9371927 | 2021-07-01 09:41:00 |     1 |       1 |
|   9371998 | 2021-07-01 09:45:00 |     0 |       1 |
|   9372121 | 2021-07-01 09:48:00 |     0 |       1 |
|   9372211 | 2021-07-01 09:51:00 |     1 |       1 |
|   9372328 | 2021-07-01 09:51:00 |     1 |       1 |
|   9372338 | 2021-07-01 09:54:00 |     0 |       1 |
|   9372362 | 2021-07-01 09:54:00 |     0 |       1 |
|   9373363 | 2021-07-01 10:25:00 |     1 |       1 |
|   9373412 | 2021-07-01 10:27:00 |     1 |       1 |
|   9373575 | 2021-07-01 10:33:00 |     1 |       1 |
|   9373598 | 2021-07-01 10:32:00 |     1 |       1 |
|   9376262 | 2021-07-01 12:01:00 |     1 |       1 |
|   9379021 | 2021-07-01 13:30:00 |     1 |       1 |
|   9381033 | 2021-07-01 14:31:00 |     1 |       1 |
|   9381165 | 2021-07-01 14:33:00 |     1 |       1 |
|   9384423 | 2021-07-01 16:11:00 |     1 |       1 |
|   9384623 | 2021-07-01 16:18:00 |     1 |       1 |
|   9384676 | 2021-07-01 16:18:00 |     1 |       1 |
|   9384712 | 2021-07-01 16:21:00 |     1 |       1 |
|   9384796 | 2021-07-01 16:22:00 |     1 |       1 |
|   9384892 | 2021-07-01 16:26:00 |     1 |       1 |
+-----------+---------------------+-------+---------+
30 rows in set (0.00 sec)
```

느낀 점

감사합니다.