In [1]:
```
import sys
print(sys.executable)
```

```
c:\Users\xrtwin\miniforge3\envs\dacon\python.exe
```

Import

In [2]:
```
import pandas as pd
import numpy as np

from sklearn.ensemble import IsolationForest
from sklearn.metrics import f1_score
from sklearn.metrics import classification_report

import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings(action='ignore')
```

Data Load

In [3]:
```
train_df = pd.read_csv('open/train.csv') # Train
train_df.head()
```

Out[3]:

	ID	V1	V2	V3	V4	V5	V6	V7	
0	3	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.24
1	4	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.37
2	6	-0.425966	0.960523	1.141109	-0.168252	0.420987	-0.029728	0.476201	0.26
3	8	-0.644269	1.417964	1.074380	-0.492199	0.948934	0.428118	1.120631	-3.80
4	9	-0.894286	0.286157	-0.113192	-0.271526	2.669599	3.721818	0.370145	0.85

5 rows × 31 columns

In [4]:
```
val_df = pd.read_csv('open/val.csv') # Validation
val_df.head()
```

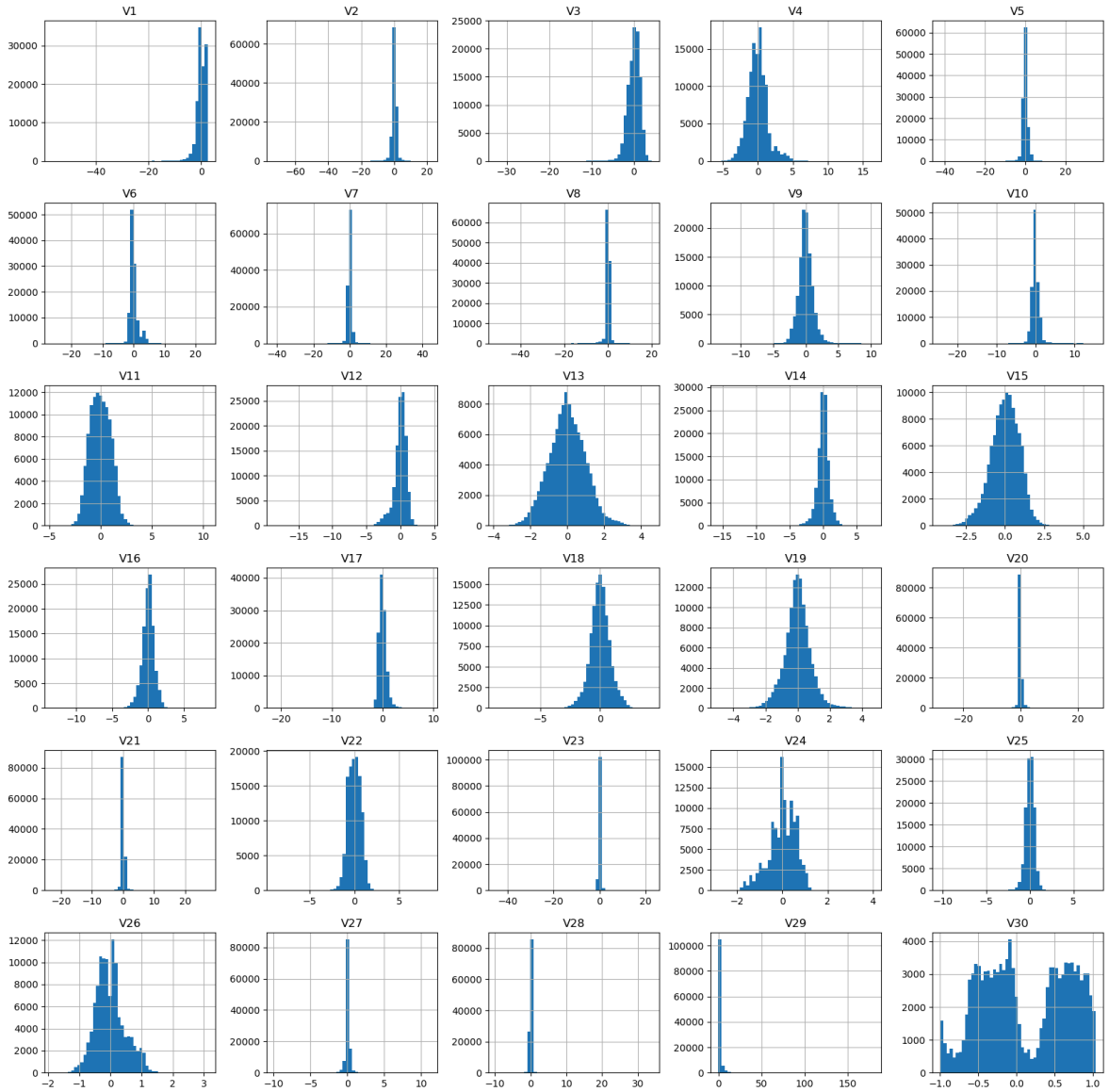
Out[4]:

	ID	V1	V2	V3	V4	V5	V6	V7	
0	10	-0.338262	1.119593	1.044367	-0.222187	0.499361	-0.246761	0.651583	0.0
1	22	0.962496	0.328461	-0.171479	2.109204	1.129566	1.696038	0.107712	0.5
2	63	1.145524	0.575068	0.194008	2.598192	-0.092210	-1.044430	0.531588	-0.2
3	69	0.927060	-0.323684	0.387585	0.544474	0.246787	1.650358	-0.427576	0.6
4	83	-3.005237	2.600138	1.483691	-2.418473	0.306326	-0.824575	2.065426	-1.8

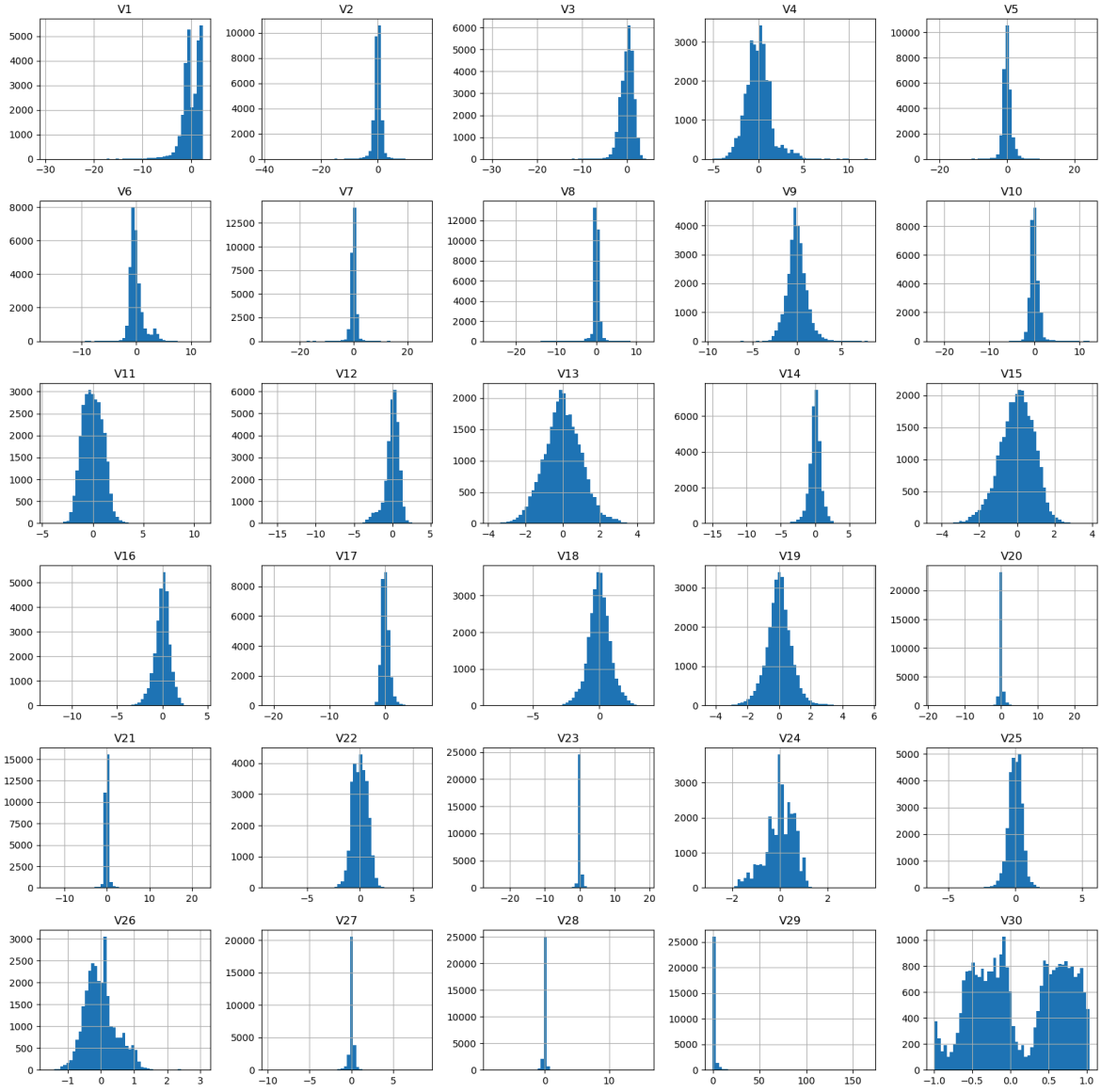
5 rows × 32 columns

Train/Validation Feature 분포 확인

In [5]:
```
train_df.drop(columns=['ID']).hist(bins = 50, figsize = (20,20))
plt.show()
```



In [6]:
```
val_df.drop(columns=['ID', 'Class']).hist(bins = 50, figsize = (20,20))
plt.show()
```



Validation set 사기 거래 비율

(*) Validation set의 사기 거래 비율이 다른 데이터집합에서도 비슷하게 발생할 것이라고 가정

In [7]:
```
val_normal, val_fraud = val_df['Class'].value_counts()
val_contamination = val_fraud / val_normal
print(f'Validation contamination : [{val_contamination}]')
```

```
Validation contamination : [0.0010551491277433877]
```

Model Define & Fit

In [8]:
```
# Train dataset은 Label이 존재하지 않음
train_x = train_df.drop(columns=['ID']) # Input Data
```

In [9]:
```
# 가설 설정 : Train dataset도 Validation dataset과 동일한 비율로 사기거래가 발생
model = IsolationForest(n_estimators=125, max_samples=len(train_x), contaminatio
model.fit(train_x)
```

Out[9]:

IsolationForest

▶ Parameters

Evaluation : Validation set

In [10]:
```
def get_pred_label(model_pred):
    # IsolationForest 모델 출력 (1:정상, -1:불량(사기)) 이므로 (0:정상, 1:불량(사
    model_pred = np.where(model_pred == 1, 0, model_pred)
    model_pred = np.where(model_pred == -1, 1, model_pred)
    return model_pred
```

In [11]:
```
val_x = val_df.drop(columns=['ID', 'Class']) # Input Data
val_y = val_df['Class'] # Label

val_pred = model.predict(val_x) # model prediction
val_pred = get_pred_label(val_pred)
val_score = f1_score(val_y, val_pred, average='macro')
print(f'Validation F1 Score : [{val_score}]')
print(classification_report(val_y, val_pred))
```

```
Validation F1 Score : [0.7030820840915222]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     28432
           1       0.41      0.40      0.41        30

    accuracy                           1.00     28462
   macro avg       0.71      0.70      0.70     28462
weighted avg       1.00      1.00      1.00     28462
```

Inference : Test set

In [12]:
```
test_df = pd.read_csv('open/test.csv') # Train
test_df.head()
```

Out[12]:

	ID	V1	V2	V3	V4	V5	V6	V7
0	AAAA0x1	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.23959
1	AAAA0x2	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.07880
2	AAAA0x5	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.59294
3	AAAA0x7	1.229658	0.141004	0.045371	1.202613	0.191881	0.272708	-0.00515
4	AAAA0xc	0.384978	0.616109	-0.874300	-0.094019	2.924584	3.317027	0.47045

5 rows × 31 columns

In [13]:
```
test_x = test_df.drop(columns=['ID'])
```

In [14]:
```
test_pred = model.predict(test_x) # model prediction
test_pred = get_pred_label(test_pred)
```

Submission

In [17]:
```
submit = pd.read_csv('open/sample_submission.csv')
submit.head()
```

Out[17]:

	ID	Class
0	AAAA0x1	1
1	AAAA0x2	1
2	AAAA0x5	1
3	AAAA0x7	1
4	AAAA0xc	1

In [18]:
```
submit['Class'] = test_pred
submit.to_csv('./submit.csv', index=False)
```

In [ ]:
```

```