

OpenWorld.ipynb

## Openworld :

### - SVM

1~4 feature를 사용

: mean, std, max, min, sum으로 요약해서 사용

정확도 : 0.9211, 0.3641

Binary가 0.9211

```
Binary Classification Accuracy: 0.9211363636363636
Binary Classification Report:
              precision    recall  f1-score   support

     -1         0.99         0.43         0.60         609
         1         0.92         1.00         0.96        3791

 accuracy                   0.92         4400
 macro avg              0.95         0.72         0.78         4400
 weighted avg           0.93         0.92         0.91         4400
```

-> 전반적으로 정확도는 높지만, 실제 **unmonitored** 중 43%만 올바르게 예측함(-1의 recall이 0.43)

SVM은 경계 기반 모델로, **margin**을 최대화하려고 하기 때문에 불균형 데이터셋을 학습시킬 때는 다수인 클래스에 더 큰 중요성을 둬. 따라서 소수 클래스에 대한 예측 정확도가 떨어지는 경향이 있음.

-1(un)은 샘플의 개수가 10000개, 1은 19000개로 거의 2배인 것이 영향을 줬을 수 있음

Multi가 0.3641

Multi-Class Classification Accuracy: 0.3640909090909091

Multi-Class Classification Report:

	precision	recall	f1-score	support
-1	0.22	0.82	0.35	609
0	0.00	0.00	0.00	40
1	0.33	0.24	0.28	46
2	0.00	0.00	0.00	43
3	0.00	0.00	0.00	47
4	0.10	0.05	0.06	43
5	0.70	0.35	0.47	40
6	0.42	0.76	0.54	29
7	0.67	0.35	0.46	52
8	0.85	0.27	0.41	41
9	0.20	0.02	0.04	41

클래스 0, 2, 3, 4는 precision과 recall이 0에 가깝거나 매우 낮음,

Precision is ill-defined and being set to 0.0 in labels with no predicted samples 경고가 뜸. 이 경고는 모델이 특정 클래스에 대해 전혀 예측을 하지 않아서 정밀도를 계산할 수 없다는 경고임. 이 클래스를 전혀 예측하지 않았다는 것은 매우 정확도가 낮다는 것을 의미함.

-1 클래스(unmonitored)의 precision은 0.22, recall은 0.82로 많은 -1 예측이 실제로는 0~94 클래스에 속하는 샘플이었음을 알 수 있음.

-> -1은 3000, 0~94는 각 200개씩의 샘플을 가진 것이 monitored에 underfitting을 하게 해서 모델이 -1을 과도하게 예측한 데에 영향을 줬을 수 있음

## 5~9 사용

정확도 : 0.8618, 0.1957

-> 1~4를 사용했을 때와 비슷한 결과

Binary 에서

=== Binary Classification (Open-World) ===

Accuracy: 86.18%

Classification Report:

	precision	recall	f1-score	support
-1	1.00	0.00	0.00	609
1	0.86	1.00	0.93	3791
accuracy			0.86	4400
macro avg	0.93	0.50	0.46	4400
weighted avg	0.88	0.86	0.80	4400

Confusion Matrix:

```
[[ 1 608]
 [ 0 3791]]
```

confusion matrix에서 -1 클래스를 잘못 1로 예측한 수가 608개, -1 클래스를 실제로 -1로 예측한 수가 1개라는 뜻.

Multi에서 Precision이 0인 클래스가 매우 많음. 대부분이 -1 클래스로 잘못

Classification Report:

	precision	recall	f1-score	support
-1	0.15	0.87	0.25	609
0	0.00	0.00	0.00	40
1	0.00	0.00	0.00	46
2	0.00	0.00	0.00	43
3	0.00	0.00	0.00	47
4	0.00	0.00	0.00	43
5	0.00	0.00	0.00	40
6	0.00	0.00	0.00	29
7	0.00	0.00	0.00	52
8	0.00	0.00	0.00	41
9	0.00	0.00	0.00	41
10	0.00	0.00	0.00	48
11	0.81	0.39	0.52	44
12	0.55	0.61	0.58	36
13	0.00	0.00	0.00	43
14	0.00	0.00	0.00	24
15	0.00	0.00	0.00	32
16	0.64	0.17	0.27	40
17	0.00	0.00	0.00	42
18	0.00	0.00	0.00	20
19	0.00	0.00	0.00	42
20	0.50	0.72	0.59	40
21	0.00	0.00	0.00	40
22	0.00	0.00	0.00	31
23	0.00	0.00	0.00	44
24	0.08	0.15	0.11	41
25	0.00	0.00	0.00	41
26	0.00	0.00	0.00	36
분류됨.	0.00	0.00	0.00	38

## - RF

### 5~9 사용

정확도 : 0.8607, 0.8636

=== Binary Classification (Monitored vs. Unmonitored) ===

Accuracy: 0.8607

Classification Report:

	precision	recall	f1-score	support
-1	0.49	0.12	0.20	609
1	0.87	0.98	0.92	3791
accuracy			0.86	4400
macro avg	0.68	0.55	0.56	4400
weighted avg	0.82	0.86	0.82	4400

Confusion Matrix:

```
[[ 75 534]
 [ 79 3712]]
```

=== Multi-Class Classification (Monitored Classes) ===

Accuracy: 0.8636

Classification Report:

	precision	recall	f1-score	support
-1	0.47	0.12	0.19	590
1	0.88	0.98	0.93	3810
accuracy			0.86	4400
macro avg	0.67	0.55	0.56	4400
weighted avg	0.82	0.86	0.83	4400

Confusion Matrix:

```
[[ 69 521]
 [ 79 3731]]
```

-> SVM에 비해 **binary**의 정확도가 낮게 나옴, 그러나 **multi**의 경우 0.86으로 더 높은 정확도를 보였음.

ClosedWorld.ipynb

## Closed World

1~4를 최빈값을 선택해서 전처리

- **RF**

X1만 썼을 땐 0.4579

1,2 썼을 땐 0.6029

**1,2,3** 썼을 땐 **0.9100**

-> **1,2,3 feature**를 썼을 때 가장 높은 정확도가 나옴.

-> 1,2,3,4,5,6,7,8,9 feature를 썼을 때 어떤 결과가 나올까?

s\_final\_categorical.ipynb

## Open world :

### - RF

1~3 리스트 데이터를 요약하고 스케일링을 적용(4는 누적 크기로 계산은 하였으나 코드의 편의상 사용하지 않음), 트리 수 200, monitored와 unmonitored의 불균형 보정을 함

```
# Step 6: 랜덤포레스트 모델 생성 및 학습
def train_and_evaluate(X_train, X_test, y_train, y_test, mode):
    print(f"\n=== {mode.capitalize()} Classification ===")
    rf_model = RandomForestClassifier(
        n_estimators=200,
        max_depth=None,
        min_samples_split=2,
        max_features='sqrt',
        random_state=42,
        class_weight='balanced'
    )
```

정확도 : 0.95, 0.65

binary에서 0.95

```
=== Binary Classification ===
Accuracy: 0.95
Classification Report:
              precision    recall  f1-score   support

     -1       0.98       0.65       0.78       882
       1       0.95       1.00       0.97      5718

 accuracy          0.95          0.95          0.95          6600
 macro avg         0.96          0.82          0.88          6600
 weighted avg         0.95          0.95          0.95          6600
```

탐지되지 않은 unmonitored가 많음을 의미

Monitored에 overfitting 되었을 수 있음

여전히 Unmonitored에는 인스턴스가 3000, monitored는 19000인 것이 영향을 줬을 가능성 있음

하지만 5~9 feature만 사용했을 때보다 binary에선 높은 정확도를 보임

Multi에서 0.65

```

=== Multi-class Classification ===
Accuracy: 0.65
Classification Report:

```

	precision	recall	f1-score	support
-1	0.51	0.55	0.53	882
0	0.63	0.55	0.59	58
1	0.53	0.30	0.38	60
2	0.86	0.89	0.87	62
3	0.53	0.57	0.55	46
4	0.57	0.78	0.66	50
5	0.88	0.83	0.85	60
6	0.85	0.90	0.88	63
7	0.53	0.60	0.56	50
8	0.68	0.67	0.67	54
9	0.60	0.54	0.57	48
10	0.65	0.55	0.60	67
11	0.70	0.61	0.65	64
12	0.78	0.89	0.83	61
13	0.58	0.34	0.43	76
14	0.56	0.42	0.48	57

클래스 2에는 0.86, 0.89, 0.87이라는 높은 점수를 갖고 있지만,

클래스 14은 0.58, 0.34, 0.43으로 낮은 점수를 갖고 있음

Feature가 너무 많아서 multi에서 이전보다 낮은 정확도가 나왔을 가능성이 있음

## - SVM

RF와 거의 동일한 조건에서 실험

정확도 : 0.72, 0.27

binary에서 0.72

```

=== Binary Classification (SVM) ===
Accuracy: 0.72
Classification Report:

```

	precision	recall	f1-score	support
-1	0.23	0.48	0.31	882
1	0.90	0.75	0.82	5718
accuracy			0.72	6600
macro avg	0.57	0.62	0.57	6600
weighted avg	0.81	0.72	0.75	6600

두 모델 모두 **monitored(1)** 데이터를 잘 분류했지만, **RF**가 더 완벽에 가까운 성능을 보임. **Unmonitored**도 **RF**가 훨씬 우수한 성능을 가짐

```
=== Multi-Class Classification (SVM) ===
Accuracy: 0.27
Classification Report:

```

	precision	recall	f1-score	support
-1	1.00	0.01	0.02	882
0	0.83	0.09	0.16	58
1	0.02	0.02	0.02	60
2	0.14	0.35	0.20	62
3	0.20	0.07	0.10	46
4	0.12	0.08	0.10	50
5	0.00	0.00	0.00	60
6	0.33	0.73	0.46	63
7	0.16	0.38	0.23	50
8	0.09	0.04	0.05	54
9	0.19	0.35	0.25	48
10	0.12	0.01	0.03	67
11	0.53	0.39	0.45	64
12	0.36	0.52	0.42	61
13	0.00	0.00	0.00	76
14	0.00	0.00	0.00	57

**SVM**은 기본적으로 경계를 만드는 모델이기 때문에 비선형적으로 데이터가 분포했을 경우와 잘못된 **feature**를 선택했을 때 성능이 떨어짐.

**RF**는 앙상블 학습을 한다는 특성이 있기 때문에 데이터 불균형과 잘못된 **feature** 선택에 상대적으로 영향을 덜 받음.

**SVM**의 정확도가 이전에 비해 많이 떨어졌다는 것은 **9개 feature**를 모두 사용할 필요는 없다는 것을 의미함

## Closed world

### - RF

정확도 : 0.67



Closed-world Experiment Performance:

Accuracy: 0.67

Classification Report:

	precision	recall	f1-score	support
0	0.71	0.56	0.63	62
1	0.41	0.25	0.31	59
2	0.71	0.91	0.80	55
3	0.53	0.76	0.62	46
4	0.70	0.74	0.72	53
5	0.79	0.79	0.79	63
6	0.74	0.94	0.83	65
7	0.71	0.58	0.64	64
8	0.71	0.68	0.70	59
68	0.76	0.54	0.63	70
69	0.62	0.50	0.55	66
70	0.92	0.95	0.94	62
71	0.58	0.63	0.61	57
72	0.65	0.57	0.61	63

클래스 70은 0.92, 0.95, 0.94로 매우 좋은 성능을 보이지만 클래스 1은 0.41, 0.25, 0.31로 성능이 낮음. 정밀도와 재현율이 모두 낮아서 클래스 1에 대해서 아예 잘못 알고 분류하는 것을 보임