

[17페이지: 아래서 1행]

데이터를 분석하고 머신러닝 모델을 구축하는 방법에 이미 익숙하다면 데이터 세트를 자유롭게 잡고 3장으로 건너뛰면 바로 MLOps에 뛰어 들 수 있다.

→

데이터를 분석하고 머신러닝 모델을 구축하는 방법에 이미 익숙하다면 데이터 세트를 자유롭게 확보하고 3장으로 넘어가면 바로 MLOps에 진입할 수 있다.

[22페이지: 아래서 3행]

모델의 학습 정확도가 떨어지고 학습 손실이 크면 높은 편향을 받게 된다. 즉, 모델이 학습 데이터에 대해 작업을 올바르게 수행하는 방법을 전혀 학습하지 못하고 있다는 뜻이다.

→

모델의 학습 정확도가 떨어지고 학습 손실이 크면 높은 편향으로 어려움을 겪게 된다. 즉, 모델이 학습 데이터에 대해 작업을 올바르게 수행하는 방법을 전혀 학습하지 못하고 있다는 뜻이다.

[32페이지: 2행]

정상 데이터를 살펴보면 이상 징후보다 오른쪽으로 훨씬 치우쳐 있다. 거래의 대부분은 100달러 이하이며, 일부 금액은 25,000달러와 같은 가치까지 올라갈 수 있다.

→

정상 데이터를 살펴보면 이상 징후보다 오른쪽으로 훨씬 치우쳐 있다. 거래의 대부분은 100달러 이하이며, 일부 금액은 25,000달러 정도까지 올라갈 수 있다.

[35페이지: 그림 1-14]

그림 1-14 데이터프레임의 Amount 칼럼에 있는 구간으로 구성된 데이터 값의 카운트 히스토그램이다. bin 수는 100이다. 즉, 히스토그램에서 각 막대의 구간은 Amount 칼럼의 데이터 범위를 bin 수로 나눈 값이다.

→

그림 1-14 데이터프레임의 Amount 칼럼에 있는 구간으로 구성된 데이터 값의 카운트 히스토그램이다. bin 수는 100이다. 즉, 히스토그램에서 각 막대의 구간은 Amount 칼럼의 데이터 범위를 bin 수로 나눈 값이다.

[40페이지: 2행]

다. 부정 거래가 포함하는 값의 범위는 v1의 값으로 확장된다. 계속해서 양에 대한 값이 v1과 어떻게 관련돼 있는지 살펴보겠다.

→

다. 부정 거래가 포함하는 값의 범위는 v1의 값으로 확장된다. 계속해서 Amount에 대한 값이 v1과 어떻게 관련돼 있는지 살펴보겠다.

[57페이지: 첫 번째 코드]

```
x_train, x_test, y_train, y_test = train_test_train(x, y, test_size=0.2,  
random_state = 2020)
```

→

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,  
random_state = 2020)
```

[57페이지: 두 번째 코드 1행]

```
normal_train, normal_test = train_test_train(normal, test_size = 0.2,  
random_state = 2020)  
anomaly_train, anomaly_test = train_test_train(anomaly, test_size = 0.2,  
random_state = 2020)
```

→

```
normal_train, normal_test = train_test_train(normal, test_size = 0.2,  
random_state = 2020)  
anomaly_train, anomaly = train_test_split(anomaly, test_size = 0.2,  
random_state = 2020)
```

[57페이지: 두 번째 코드 3행]

```
normal_train, normal_test = train_test_train(normal, test_size = 0.2,  
random_state = 2020)  
anomaly_train, anomaly_test = train_test_train(anomaly, test_size = 0.2,  
random_state = 2020)
```

→

```
normal_train, normal_test = train_test_train(normal, test_size = 0.2,  
random_state = 2020)  
anomaly_train, anomaly = train_test_split(anomaly, test_size = 0.2,  
random_state = 2020)
```

[57페이지: 두 번째 코드 3행]

```
normal_train, normal_test = train_test_train(normal, test_size = 0.2,  
random_state = 2020)  
anomaly_train, anomaly_test = train_test_train(anomaly, test_size = 0.2,  
random_state = 2020)
```

→

```
normal_train, normal_test = train_test_train(normal, test_size = 0.2,  
random_state = 2020)  
anomaly_train, anomaly = train_test_split(anomaly, test_size = 0.2,  
random_state = 2020)
```

[57페이지: 두 번째 코드 3행]

```
normal_train, normal_test = train_test_train(normal, test_size = 0.2,  
random_state = 2020)  
anomaly_train, anomaly_test = train_test_train(anomaly, test_size = 0.2,  
random_state = 2020)
```

→

```
normal_train, normal_test = train_test_train(normal, test_size = 0.2,  
random_state = 2020)  
anomaly_train, anomaly = train_test_split(anomaly, test_size = 0.2,  
random_state = 2020)
```

[58페이지: 첫 번째 코드 1행]

```
normal_train, normal_train = train_test_train(normal_train, test_size = 0.25,  
random_state = 2020)  
anomaly_train, anomaly_train = train_test_train(anomaly_train, test_size =  
0.25, random_state = 2020)
```

→

```
normal_validate, normal_validate = train_test_split(normal_train, test_size  
= 0.25, random_state = 2020)  
anomaly_train, anomaly_train = train_test_split(anomaly_validate, test_size  
= 0.25, random_state = 2020)
```

[58페이지: 첫 번째 코드 1행]

```
normal_train, normal_train = train_test_train(normal_train, test_size = 0.25,  
random_state = 2020)  
anomaly_train, anomaly_train = train_test_train(anomaly_train, test_size =  
0.25, random_state = 2020)
```

→

```
normal_validate, normal_validate = train_test_split(normal_train, test_size  
= 0.25, random_state = 2020)  
anomaly_train, anomaly_train = train_test_split(anomaly_validate, test_size  
= 0.25, random_state = 2020)
```

[58페이지: 첫 번째 코드 3행]

```
normal_train, normal_train = train_test_train(normal_train, test_size = 0.25,  
random_state = 2020)  
anomaly_train, anomaly_train = train_test_train(anomaly_train, test_size =  
0.25, random_state = 2020)
```

→

```
normal_validate, normal_validate = train_test_split(normal_train, test_size  
= 0.25, random_state = 2020)  
anomaly_train, anomaly_train = train_test_split(anomaly_validate, test_size  
= 0.25, random_state = 2020)
```

[58페이지: 첫 번째 코드 3행]

```
normal_train, normal_train = train_test_train(normal_train, test_size = 0.25,  
random_state = 2020)  
anomaly_train, anomaly_train = train_test_train(anomaly_train, test_size =  
0.25, random_state = 2020)
```

→

```
normal_validate, normal_validate = train_test_split(normal_train, test_size  
= 0.25, random_state = 2020)  
anomaly_train, anomaly_train = train_test_split(anomaly_validate, test_size  
= 0.25, random_state = 2020)
```

[58페이지: 첫 번째 코드 3행]

```
normal_train, normal_train = train_test_train(normal_train, test_size = 0.25,  
random_state = 2020)  
anomaly_train, anomaly_train = train_test_train(anomaly_train, test_size =  
0.25, random_state = 2020)
```

→

```
normal_validate, normal_validate = train_test_split(normal_train, test_size  
= 0.25, random_state = 2020)  
anomaly_train, anomaly_train = train_test_split(anomaly_validate, test_size  
= 0.25, random_state = 2020)
```

[59페이지: 1행]

마지막으로 레이블을 직접 제공하면 정상 트랜잭션과 비정상 트랜잭션을 구성하는 요소를 하는 방법을 모델에 가르치는 데 방해가 될 수 있으므로 X 집합의 Class 열을 삭제해야 한다.

→

마지막으로, 레이블을 직접 지정하는 경우 정상 및 사기 거래를 구성하는 항목을 학습하는 방법을 모델에 가르치기 위해 x 세트에서 Class 열을 삭제해야 한다.

[60페이지: 5행]

조금 더 상세히 말하면 모델은 비용함수를 최적화하는 데 어려움을 겪을 것이며, 만약 그렇게 할 수 있다면 수렴하기 위해 더 많은 단계를 밟을 수 있다.

→

조금 더 상세히 말하면 모델은 비용함수를 최적화하는 데 어려움을 겪을 것이며, **가능하다면** 수렴하기 위해 더 많은 단계를 밟을 수 있다.

[127페이지: 첫 번째 코드 3행]

```
normal = df[df.Class == 0].sample(frac=0.5,  
random_state=2020).reset_index(drop=True)  
anomaly = df[df.Class == 1]
```

→

```
normal = df[df.Class == 0].sample(frac=0.5,  
random_state=2020).reset_index(drop=True)  
anomaly = df[df.Class == 1]
```