

## 과대적합(Overfitting)/과소적합(Underfitting)이란?

---

### ✓ 과대적합(overfitting)이란?

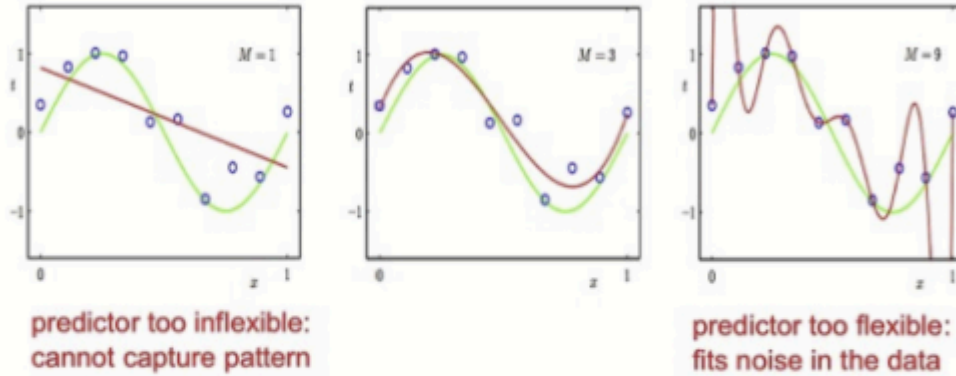
- 머신러닝 모델을 학습할 때 학습 데이터셋에 지나치게 최적화하여 발생하는 문제
  - 모델을 지나치게 복잡하게 학습하여 학습 데이터셋에서는 모델 성능이 높게 나타나지만 정작 새로운 데이터가 주어졌을 때 정확한 예측/분류를 수행하지 못함
- 발생 원인
  - 데이터 세트 내 데이터가 충분하지 못한 경우
  - 데이터 세트 내 분산이 크거나 노이즈가 큰 경우
  - 모델의 복잡도가 큰 경우
  - 과도하게 큰 epoch로 학습한 경우
- 해결 방법
  - 데이터 양 늘리기
    - 데이터 양이 적어서 해당 데이터의 특징 패턴, 노이즈까지 학습해버리기 때문에 데이터 양을 늘려야 모델은 일반적인 패턴을 학습하여 과대적합을 방지할 수 있다.
  - 모델의 복잡도 줄이기
  - Dropout 사용하기

## ✓ 과소적합(underfitting)이란?

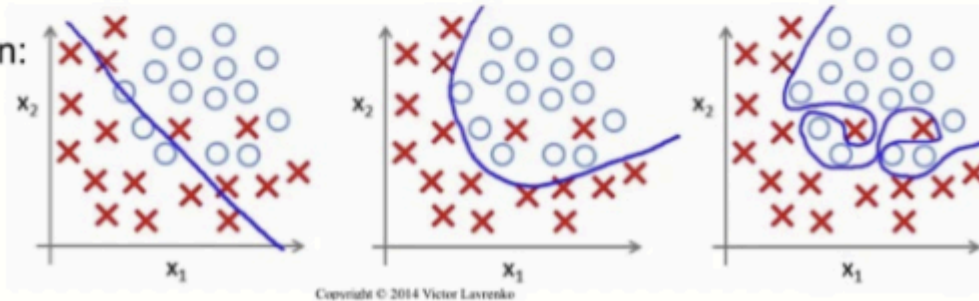
- 머신러닝 모델이 충분히 복잡하지 않아(최적화가 제대로 수행되지 않아) 학습 데이터의 구조/패턴을 정확히 반영하지 못하는 문제
- 발생 원인
  - 충분한 샘플들이 주어지지 않았을 때
  - 데이터의 양이 적을 때
- 해결 방법
  - 파라미터가 더 많은 복잡한 모델을 선택한다.
  - 더 좋은 특성들을 제공한다(Feature Engineering)
  - 규제 하이퍼파라미터를 감소시키는 등의 방법으로 모델의 제약을 줄인다.

# Under- and Over-fitting

Regression:



Classification:



```

1 #시각화를 위해 2개의 feature와 3개의 클래스를 가진 임의의 샘플을 만들
2 #이때, feature의 갯수를 2개로 한 것은 2차원 평면상에 표현하기 위함
3 #3개의 Sample Class는 각 클래스들의 색깔을 다르게하여 표현
4 #https://jaylala.tistory.com/entry/머신러닝-with-Python-결정-트리Decision-Tree-22-과적합Over-fitting
5
6 from sklearn.datasets import make_classification
7 import matplotlib.pyplot as plt
8 import numpy as np
9 %matplotlib inline
10 plt.title("3 Class values with 2 Features Sample data creation")

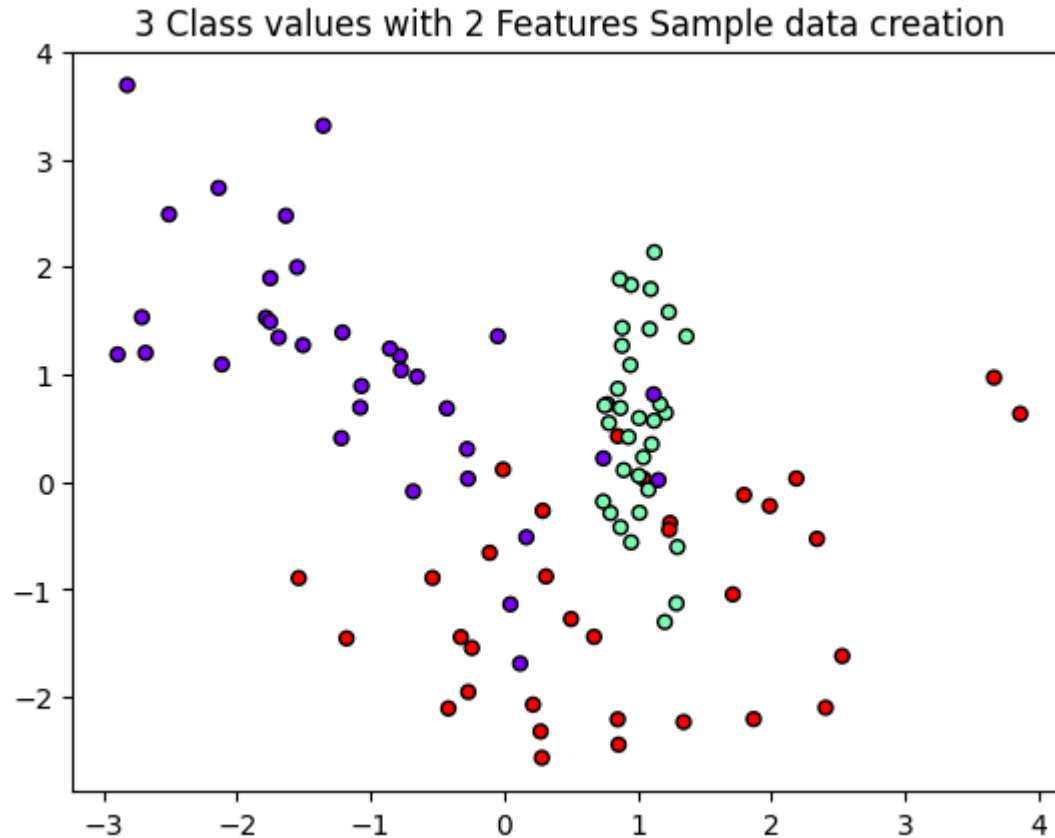
```

```

11
12 # 2차원 시각화를 위해서 feature는 2개, 결정값 클래스는 3가지 유형의 classification 샘플 데이터 생성.
13 X_features, y_labels=make_classification(n_features=2, n_redundant=0, n_informative=2, n_classes=3,n_clusters_per_class=1,random_state=0)
14
15 # plot 형태로 2개의 feature로 2차원 좌표 시각화, 각 클래스값은 다른 색깔로 표시됨.
16 plt.scatter(X_features[:,0], X_features[:, 1], marker='o', c=y_labels, s=25, cmap='rainbow', edgecolor='k')

```

 <matplotlib.collections.PathCollection at 0x7ce703bdd3d0>




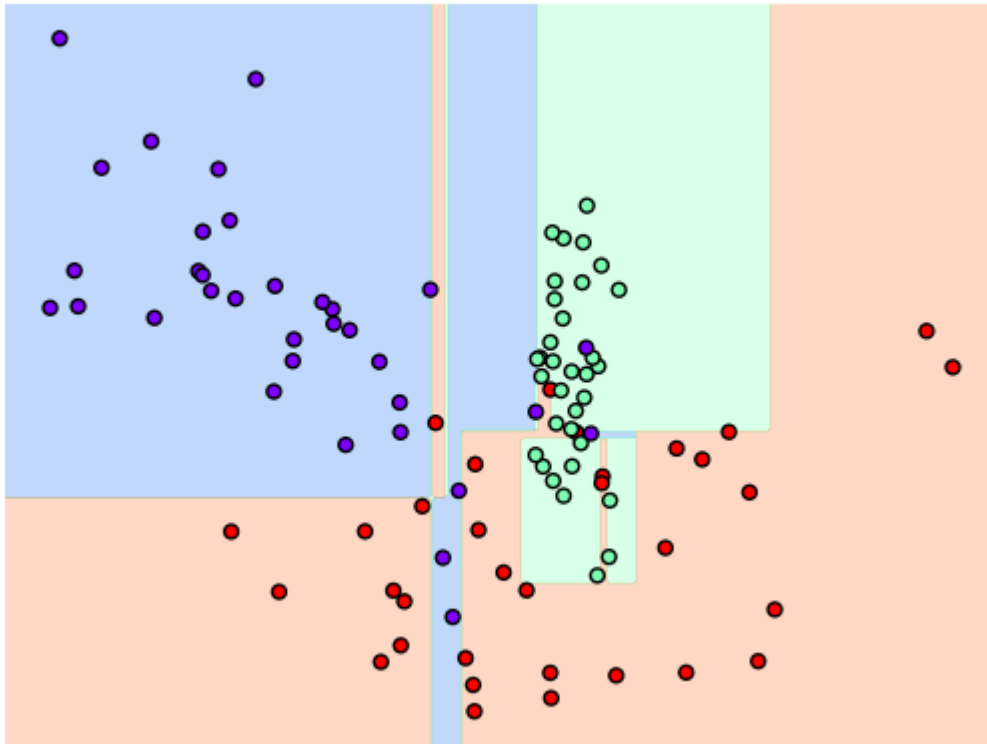
```

1 #해당 데이터를 기반으로 결정트리 모델을 만들고 결정트리 모델이 분류해낸 결과를 시각화import numpy as np
2 # Classifier의 Decision Boundary를 시각화 하는 함수
3
4 def visualize_boundary(model, X, y):
5     fig,ax=plt.subplots()

```

```
6 # 학습 데이터 scatter plot으로 나타내기
7 ax.scatter(X[:, 0], X[:, 1], c=y, s=25, cmap='rainbow', edgecolor='k', clim=(y.min(), y.max()), zorder=3)
8 ax.axis('tight')
9 ax.axis('off')
10 xlim_start, xlim_end=ax.get_xlim()
11 ylim_start, ylim_end=ax.get_ylim()
12 # 호출 파라미터로 들어온 training 데이터로 model 학습 .
13 model.fit(X, y)
14 # meshgrid 형태인 모든 좌표값으로 예측 수행.
15 xx, yy=np.meshgrid(np.linspace(xlim_start,xlim_end,num=200),np.linspace(ylim_start,ylim_end, num=200))
16 Z=model.predict(np.c_[xx.ravel(),yy.ravel()]).reshape(xx.shape)
17 # contourf() 를 이용하여 class boundary 를 visualization 수행.
18 n_classes=len(np.unique(y))
19 contours=ax.contourf(xx, yy, Z, alpha=0.3,
20 levels=np.arange(n_classes+1)-0.5, cmap='rainbow', clim=(y.min(), y.max()), zorder=1)
21 from sklearn.tree import DecisionTreeClassifier
22 # 특정한 트리 생성 제약없는 결정 트리의 Decsion Boundary 시각화.
23 dt_clf=DecisionTreeClassifier().fit(X_features,y_labels)
24 visualize_boundary(dt_clf,X_features,y_labels)
25
26 #파란색 클래스는 파란색 면적에 / 빨간색 클래스는 빨간색 면적에 / 초록색 클래스는 초록색 면적에
27 #구분이 되도록 결정트리 기본 데이터를 분류해낸 것을 확인할 수 있음
28 #이 중 각 색깔 클래스의 대부분이 속해있는 영역에 일부 다른 색깔들이 섞여있는 것을 볼 수 있는데,
29 #기본 결정트리 모델은 특정 제약조건을 설정해주지 않았기 때문에 하나하나 세세하게 분류를 해버린 것을 확인할 수 있음
30 #이렇게 복잡한 모델은 학습 데이터 세트의 특성과 약간만 다른 형태의 데이터 세트를 예측하려고 할때
31 #예측의 정확도가 현저히 떨어지게 되는 "과적합(Overfitting)" 문제가 생기게 됨
```

 <ipython-input-2-74a737c10e27>:19: UserWarning: The following kwargs were not used by contour: 'clim'  
contours=ax.contourf(xx, yy, Z, alpha=0.3,



## ✓ <<<참조자료 사이트>>>

1. [AI 훈련용 빅데이터 2026년 고갈...문제점과 대책은](#)
2. ["학습 데이터 부족" AI개발에 닥친 난관... 차세대 모델 개발 지연](#)
3. [AI 데이터 고갈 위기-"2년 후 AI 성장 멈출 수도"...데이터 절벽 '경고'](#)
4. [Undersampling과 Oversampling이란?](#)
5. [불균형 데이터\(imbalanced data\) 처리를 위한 샘플링 기법](#)

6. [샘플링 편향이라는 문제](#)

7. [데이터 품질\(Data Quality\)이란?](#)

8. [데이터에 제값 매기는 데이터 품질 관리](#)