

정 규 세 션 4 주 차

ToBig's 10기 이준걸

Machine-Learning Performance Tuning

Feature Engineering, Imbalanced Dataset,
Cloud Service

Content

Unit 01 | Feature Engineering *特征 select*

Unit 02 | Imbalanced Dataset *不平衡数据集*

Unit 03 | Cloud Service

Unit 01 | Feature Engineering

좋은 모델링을 제외하고 성능을
어떻게 올릴 것인가?

Unit 01 | Feature Engineering

성능을 올리는 꿀팁

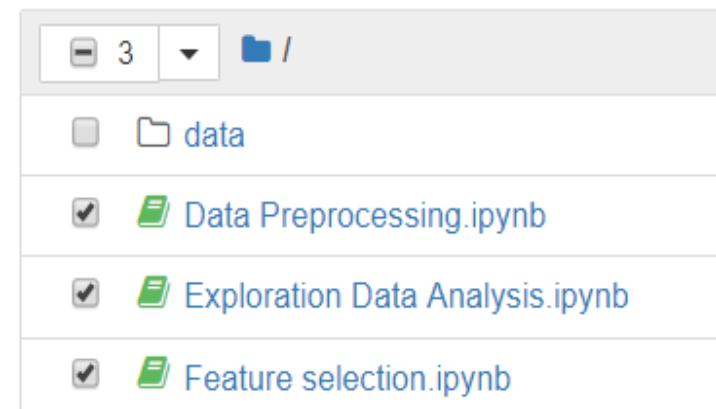
1. 모델을 변경한다.
2. 데이터를 추가적으로 확보한다
3. 엄청난 도메인을 활용해 전처리를 한다.
4. 좋은 Feature를 선택한다.
5. 좋은 Hyper parameter들을 찾아낸다.



Unit 01 | Feature Engineering

이렇게 하려면 어떻게?

1. 가지고 있는 데이터로 좋은 실험설계를 한다.(Train / Validation / Test) – 이때,
Validation의 성능이 Test의 성능과 비례하는지가 제일 중요! (Sampling issue)
2. Computing Power(현질이 어려우면 Colab을 쓰자)
3. 자동화된 코드(Data Preprocessing → Feature Selection
→ Modeling and Experiment)



Unit 01 | Feature Engineering

Feature
Feature Selection
Feature Engineering

Unit 01 | Feature Engineering

1. Feature : 데이터의 특징을 나타내는 값. – 통계학적인 말로 독립변수 or 설명변수
2. Feature Selection: 말그대로 Feature중에 쓸만한 놈들만 고르는 것
3. Feature Engineering: 새로운 Feature를 만들고 거기서 가장 적합한 놈들을 찾아내는 것! Selection보다 넓은 개념.

Unit 01 | Feature Engineering

Feature Engineering

Generation

1. Binarization, Quantization
2. Scaling(Normalization)
3. Interaction Features
4. Log Transformation
5. Dimension Reduction
6. Clustering

↳ Kmeans, PCA
PCA는 특성선택

Selection

1. Univariate Statistics
2. Model-based Selection
3. Iterative Feature Selection
4. Feature Removal

Unit 01 | Feature Engineering

Feature Generation

Unit 01 | Feature Engineering

3. Interaction Features

- 기존 Feature들의 곱셈의 조합으로 새로운 Feature를 생성
- Data에 대한 도메인이 필요
- 회귀분석에서 Polynomial Method와 동일.

이제까지는
만약

```
import numpy as np
from sklearn.preprocessing import PolynomialFeatures
```

```
df = np.array([[0,1,2], [2,3,4], [0,1,1]])
```

```
array([[0, 1, 2],
       [2, 3, 4],
       [0, 1, 1]])
```

```
poly_df = PolynomialFeatures(degree=2)
poly_df.fit_transform(df)
```

```
array([[ 1.,  0.,  1.,  2.,  0.,  0.,  0.,  1.,  2.,  4.],
       [ 1.,  2.,  3.,  4.,  4.,  6.,  8.,  9., 12., 16.],
       [ 1.,  0.,  1.,  1.,  0.,  0.,  0.,  1.,  1.,  1.]])
```

Unit 01 | Feature Engineering

Interaction Features 실습 켜주세요!

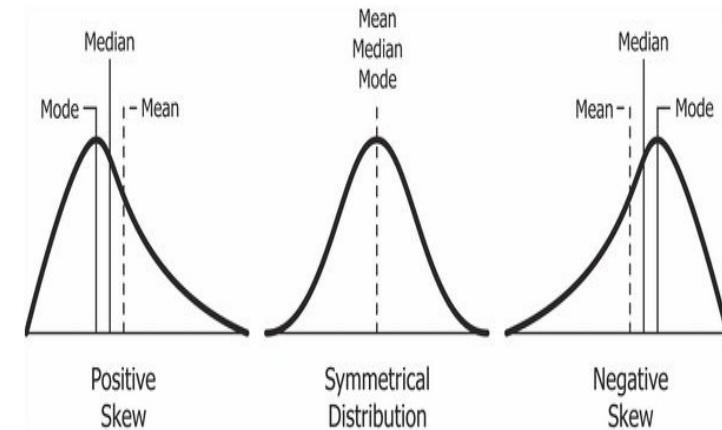
Unit 01 | Feature Engineering

4. Log Transformation

- 데이터의 분포가 극단적으로 치우쳐져 있을 때
- 선형모델은 데이터가 정규분포일 때 성능이 좋음.
- 현실에서 대부분의 데이터는 치우쳐져 있다.
- Box-Cox Transformation(log or exp를 통해 정규분포로 만들어주자)

clustering을 위한
정규분포화

$\log(x+1)$ 에서
0 증가



Unit 01 | Feature Engineering

Log Transformation
실습 켜주세요!

Unit 01 | Feature Engineering

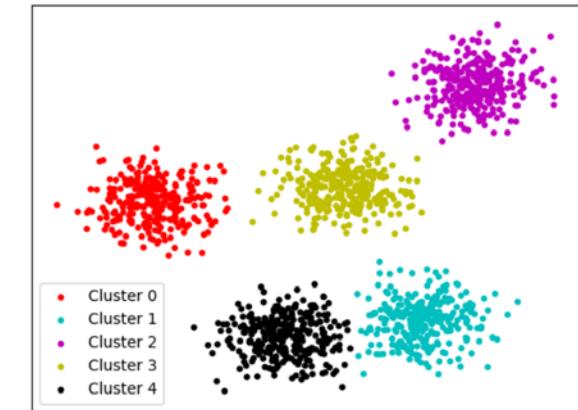
5. Dimension Reduction

- 기존 Feature 개수가 매우 많을 때
- 적절한 알고리즘을 사용해서 Feature의 공간을 축소
- PCA, t-SNE, Embedding(Word2Vec)
- 성능이 크게 좋아지진 않으나, 확실한건 시간은 매우 단축됨.
- 가끔씩 파생변수로도 사용했을 때 운좋게 좋아지는 경우 몇번 봄.


Unit 01 | Feature Engineering

6. Clustering

- 생성된 labeling값들을 Feature로 사용
- 도메인 지식이 많을 때 유용하다.
- K-means등 적절한 Clustering 기법을 사용



Unit 01 | Feature Engineering

Feature Selection

Unit 01 | Feature Engineering

Feature Selection

- 모든 Feature들이 반드시 다 필요하지 않음
- 특히 어떤 Feature는 오히려 성능을 떨어뜨림.
- 너무 많은 Feature는 또한 차원의 저주를 야기함.
- 모델에 따라서 필요한 Feature를 선택
- 필요없는 Feature를 제거 -> 속도와 성능을 향상시키는 일석이조 효과
- 무수히 많은 방법들이 있음.

Unit 01 | Feature Engineering

1. Univariate Feature Selection

- 통계 모델을 기반으로 한 최적의 Feature를 선택
- Chi Square, T-test, ANOVA 등의 통계 모델을 사용.
- Y값과 하나의 Feature간의 통계적 유의성을 파악
- 주로 선형 모델에서 유용하게 쓰임.
- 빠르게 사용할 수 있는 Feature Selection 기법.

Unit 01 | Feature Engineering

Chi Square test – 범주형 Feature

인종	내세의 믿음		total
	예	아니오	
백인	621	239	860
흑인	89	42	131
total	710	281	991

※ 자료출처 : 범주형 자료분석 개론(자유아카데미)

Unit 01 | Feature Engineering

Chi Square test – 범주형 Feature

통계량	자유도	값	Prob
카이제곱	1	1.0205	0.3124
우도비 카이제곱	1	0.9995	0.3174
연속성 수정 카이제곱	1	0.8211	0.3649
Mantel-Haenszel 카이제곱	1	1.0195	0.3126
파이 계수		0.0321	
무발성 계수		0.0321	
크레머의 V		0.0321	

귀무가설 : 내세에 대한 믿음은 인종에 따라 차이가 없다.

대립가설 : 내세에 대한 믿음은 인종에 따라 차이가 있다.

→ P-value가 0.3124로 유의수준 0.05보다 크므로 귀무가설을 기각 할 수 없다. 따라서 내세에 대한 믿음은 인종에 따라 차이가 없다고 할 수 있다.

즉 종교개혁이 아니라.

Unit 01 | Feature Engineering

T test – 연속형 Feature

Table 5-2 Catalyst Yield Data (Percent) Example 5-4

Observation Number	Catalyst 1	Catalyst 2
1	91.50	89.19
2	94.18	90.95
3	92.18	90.46
4	95.39	93.21
5	91.79	97.19
6	89.07	97.04
7	94.72	91.07
8	89.21	92.75

$\bar{x}_1 = 92.255$	$\bar{x}_2 = 92.733$
$s_1 = 2.39$	$s_2 = 2.98$
$n_1 = 8$	$n_2 = 8$

X

평균 차이가 커

2群 분석 가능

Unit 01 | Feature Engineering

T test – 연속형 Feature

Two Sample t-test

data: a and b

t = -0.35359, df = 14, p-value = 0.7289

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-3.373886 2.418886

sample estimates:

mean of x mean of y

92.2550 92.7325

귀무가설 : 두 Feature의 평균 차이가 없다.

대립가설 : 두 Feature의 평균 차이가 있다.

-> P-value가 0.7289로 유의수준 0.05보다 크므로 귀무가설을 기

각 할 수 없다. 따라서 두 Feature는 차이가 없다.

Unit 01 | Feature Engineering

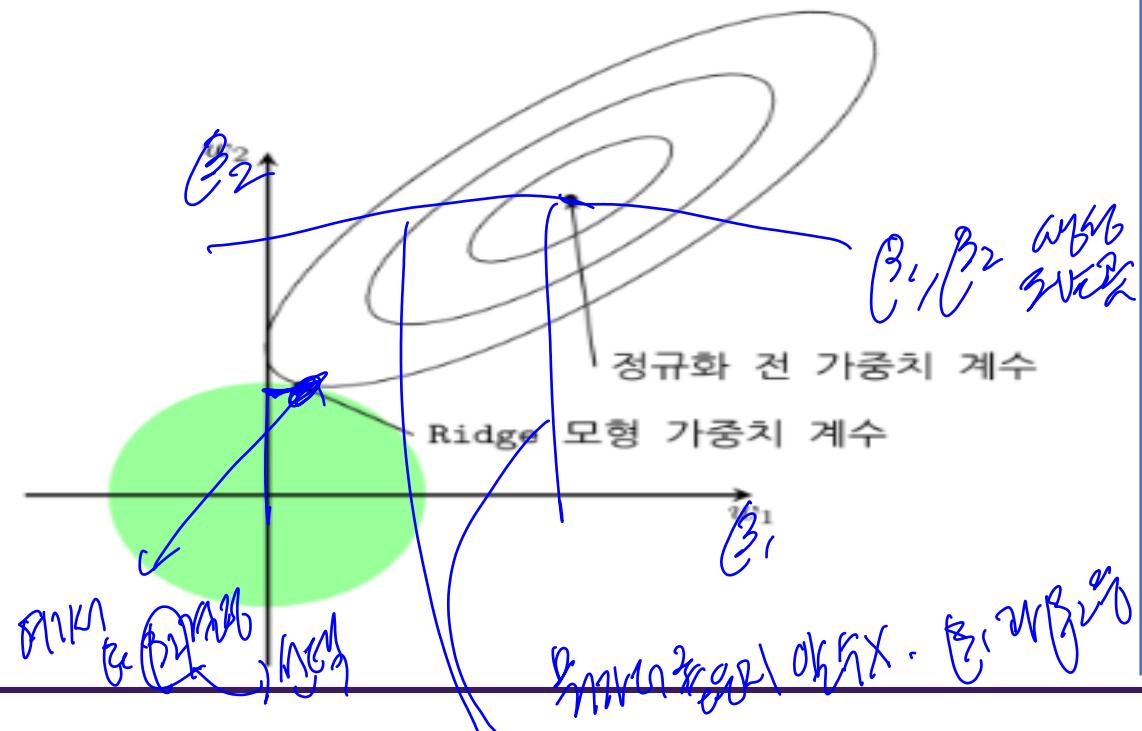
2. Model based feature selection

- 몇몇 모델은 적절한 Feature를 찾아가며 학습을 함.
- Ex) Ridge, Lasso, Tree-Based Model
- 다른 모델의 Feature 선택의 전처리 단계로 활용 가능하다.
- 한번에 모든 Feature를 고려함.
- Tree-Based Model 은 모두 이런 특징을 갖고 있다.

Unit 01 | Feature Engineering

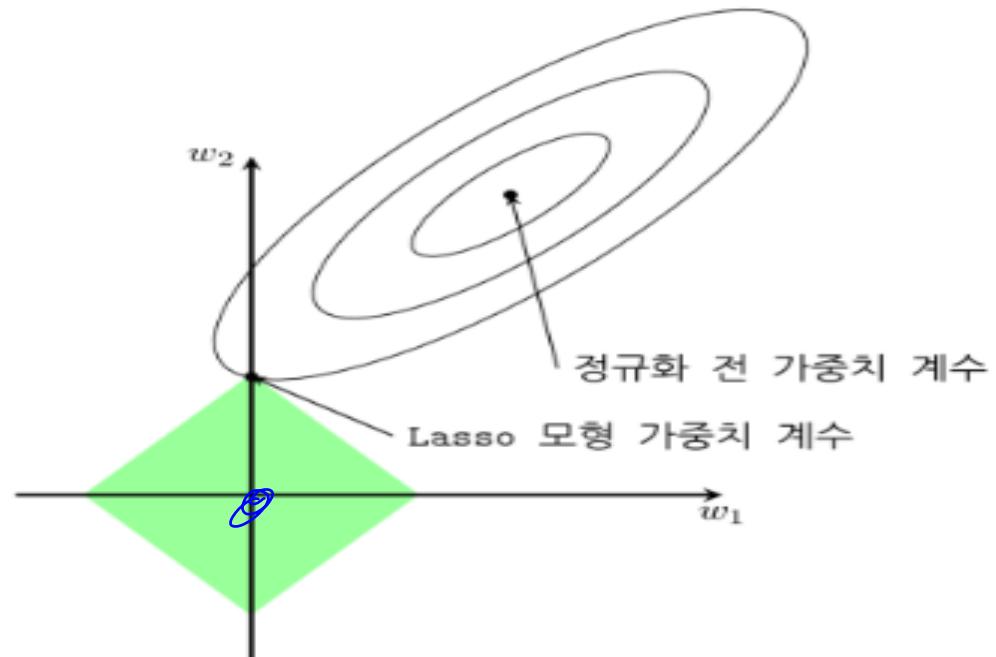
Ridge(L2)

$$w = \arg \min_w \left(\sum_{i=1}^N e_i^2 + \lambda \sum_{j=1}^M w_j^2 \right)$$



Lasso(L1)

$$w = \arg \min_w \left(\sum_{i=1}^N e_i^2 + \lambda \sum_{j=1}^M |w_j| \right)$$



Unit 01 | Feature Engineering

3. Iterative Feature Selection

Backward

- 반복적으로 feature의 수를 조절하여 최적 feature를 선택
- N개의 Feature가 있다고 하면 1개 \rightarrow N개(Forward),
N개 \rightarrow 1개(Backward) : **Recursive Feature Elimination(RFE)**
- 회귀분석에서 변수 선택방법과 유사하나, Tree 계열 모델을 사용하여
feature importance를 사용
- 데이터에 대한 도메인이 없을 때 사용하기 용이함.

Unit 01 | Feature Engineering

sklearn.feature_selection.RFE

Parameters: **estimator** : object**어떤 모델을 쓸건지?**

A supervised learning estimator with a `fit` method that provides information about feature importance either through a `coef_` attribute or through a `feature_importances_` attribute.

Linear Regression**DT기반 모델****n_features_to_select** : int or None (default=None)**몇 개 Selection 할 것인지?**

The number of features to select. If None, half of the features are selected.

step : int or float, optional (default=1)

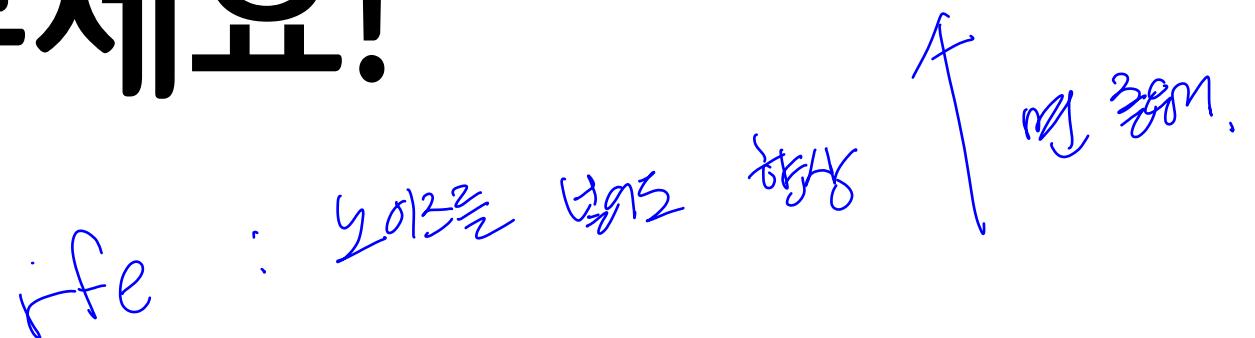
If greater than or equal to 1, then step corresponds to the (integer) number of features to remove at each iteration. If within (0.0, 1.0), then step corresponds to the percentage (rounded down) of features to remove at each iteration.

verbose : int, default=0

Controls verbosity of output.

Unit 01 | Feature Engineering

Feature Selection 실습 켜주세요!

ife : 

Unit 01 | Feature Engineering

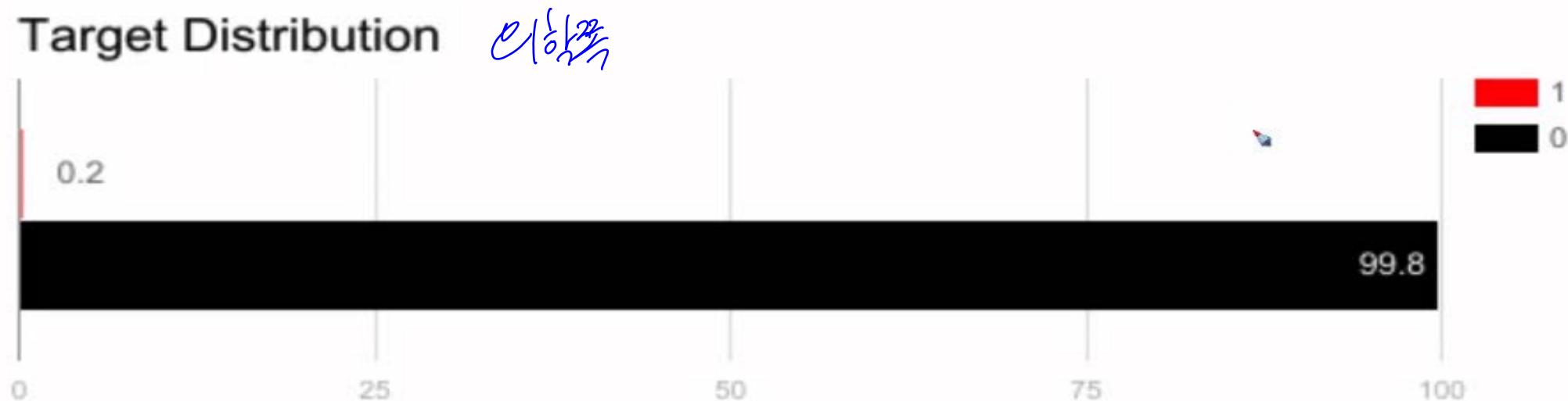
Unit 02 | Imbalanced Dataset

Unit 03 | Cloud Service

Unit 02 | Imbalanced Dataset

What is Imbalanced Dataset?

-> Class가 한쪽으로 치우쳐진 데이터 셋



현실에서 대부분의 Dataset은 imbalanced dataset이다.

Unit 02 | Imbalanced Dataset

How to handle imbalanced dataset?

- 적절한 Performance Metric을 선정(Accuracy X)
→ Precision, Recall, AUC가 적절
- 적절한 Training Dataset의 Resampling
- Resampling : Oversampling, Under Sampling, Data augmentation

과수는 데이터
만족

이상한 선택.

증강학습, 증강학습
다른 학습방법

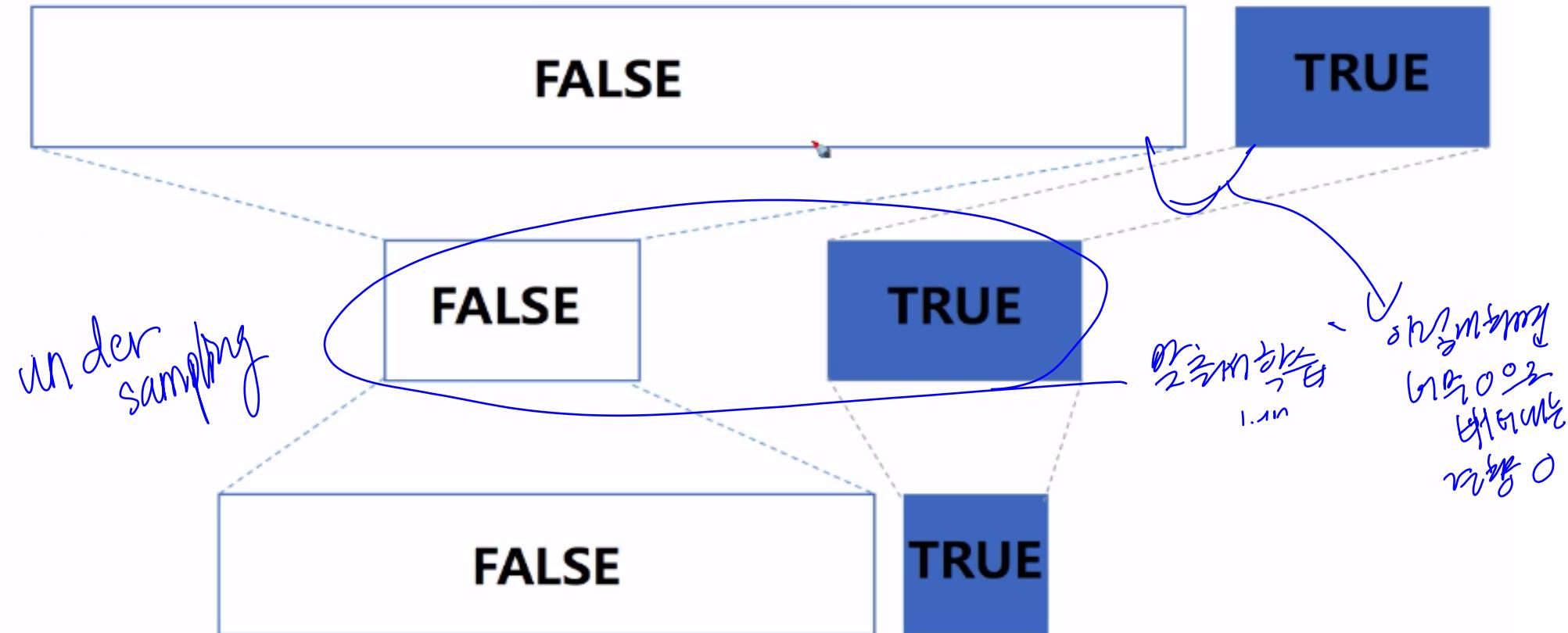
Unit 02 | Imbalanced Dataset

Dataset Resampling

original dataset



training dataset

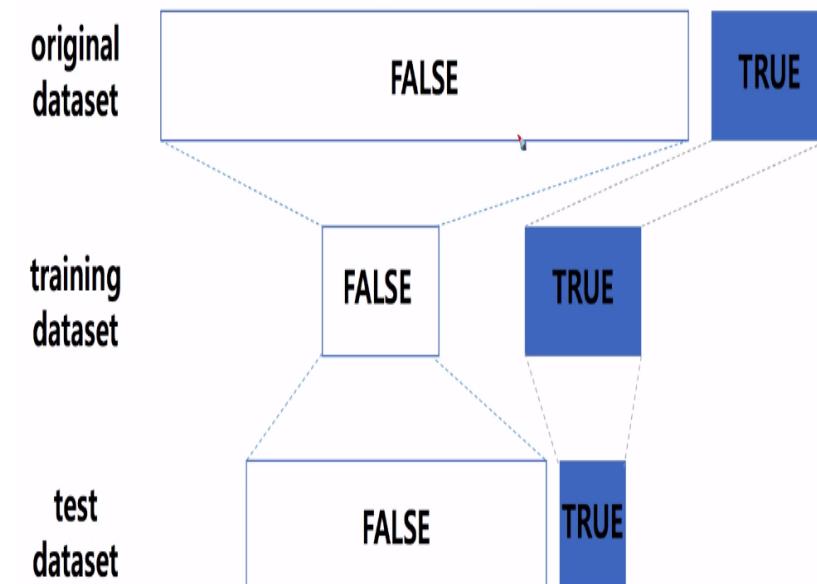


test dataset

Unit 02 | Imbalanced Dataset

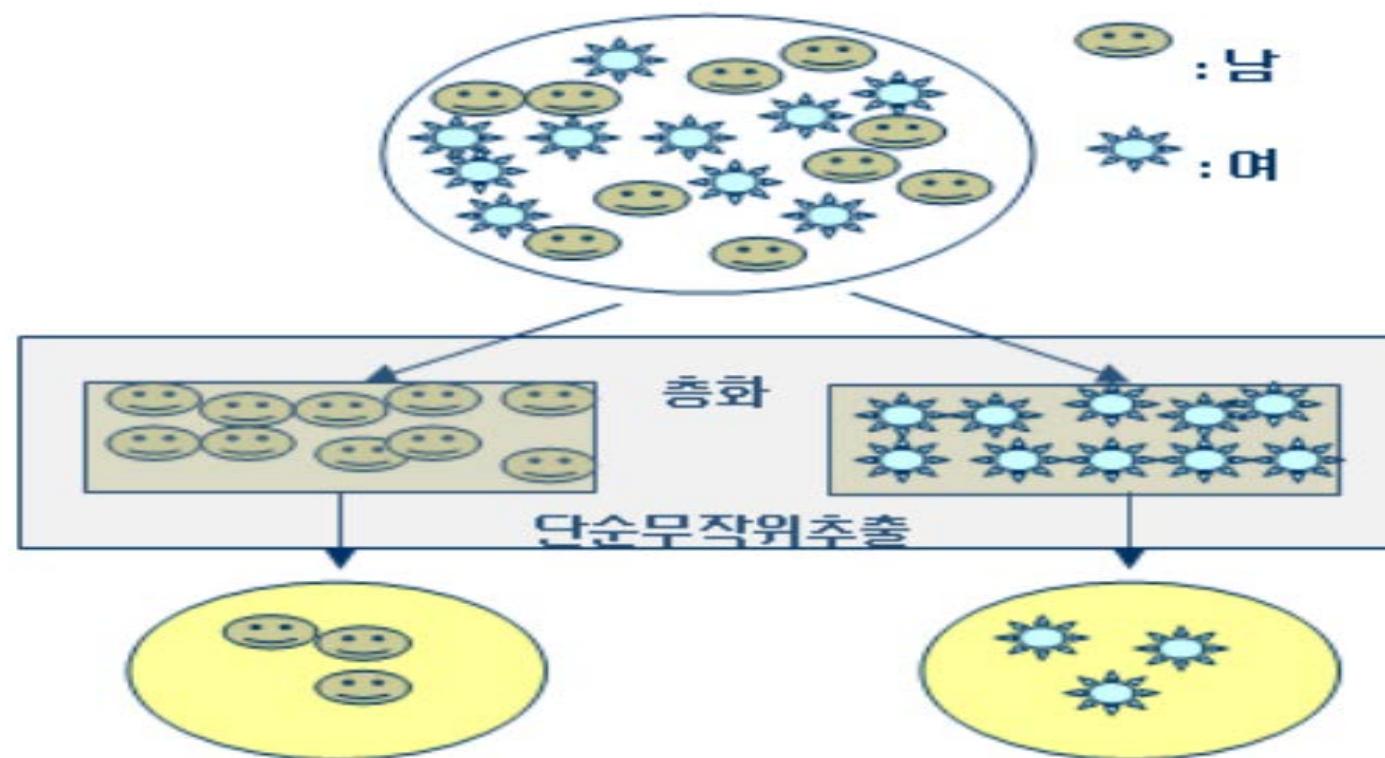
Dataset Resampling

- 1. Imbalanced class가 충분히 많다면
→ Under Sampling (False 줄임)
- 2. Imbalanced class가 부족하다면
→ Over Sampling (True를 늘림) *oversampling*



Unit 02 | Imbalanced Dataset

Stratified Resampling



Class의 개수와 비례하게
Sample을 뽑아보자!

Unit 02 | Imbalanced Dataset

Imbalanced data handling process

1. 전체 dataset에서 Test, **Dev set**으로 Stratified Resampling
(봉리오)
 2. Dev set으로 1. Under Sampling OR 2. Over Sampling → *(봉리오)*
 3. Model Training
 4. Test set으로 검증하기
- (봉리오 but Test는 봉리오 아님)*

Unit 02 | Imbalanced Dataset

Under Sampling

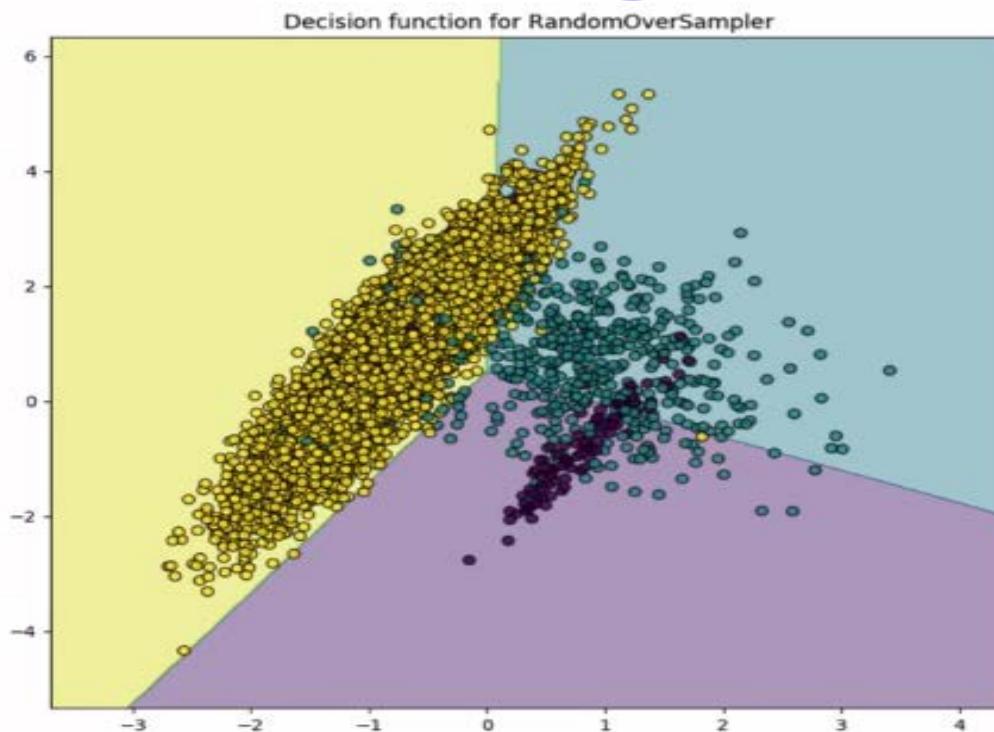
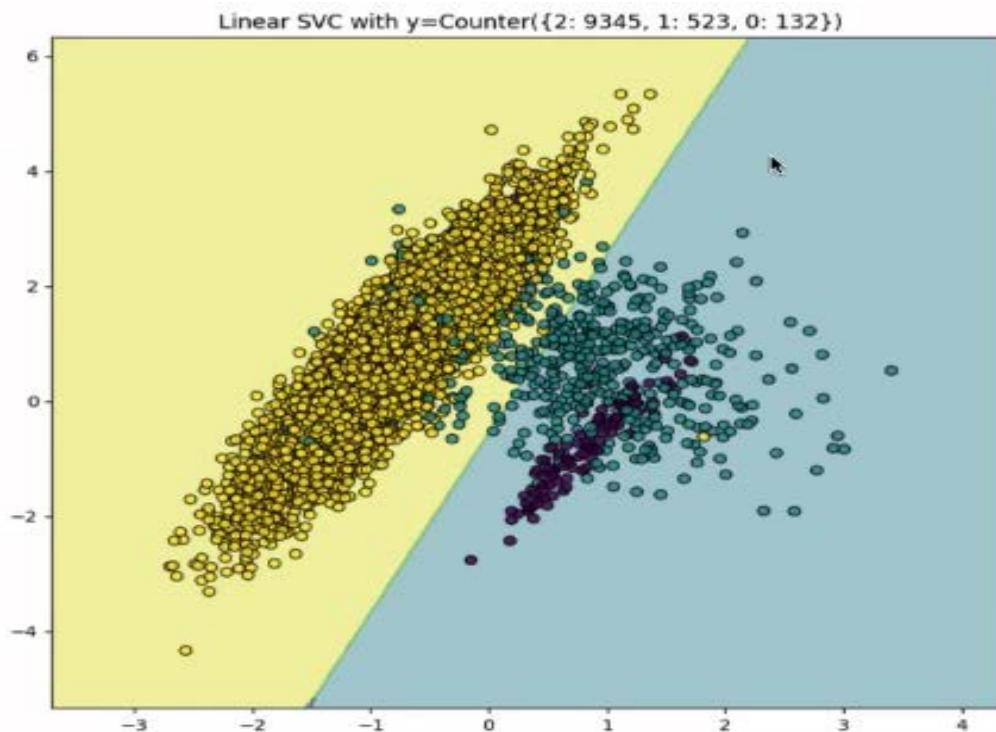
1. Random – 적은 Class를 기준으로 Random하게 선택
2. Neamiss – Heuristics based on NN algorithm
3. AllKNN – 자기 Class내에서 가장 가까운 데이터만 남김
*Class 내에서
가장 가까운
데이터만 남김*
4. Instance Hardness Threshold : 모델을 사용해서 Under Sampling
해당 모델을 통해 Class로 뱉어내는데 사용하는 확률을 기반으로
Sample을 선택

Unit 02 | Imbalanced Dataset

Over Sampling

복사

1. Random – 현재 가능한 데이터에서 일부를 복사



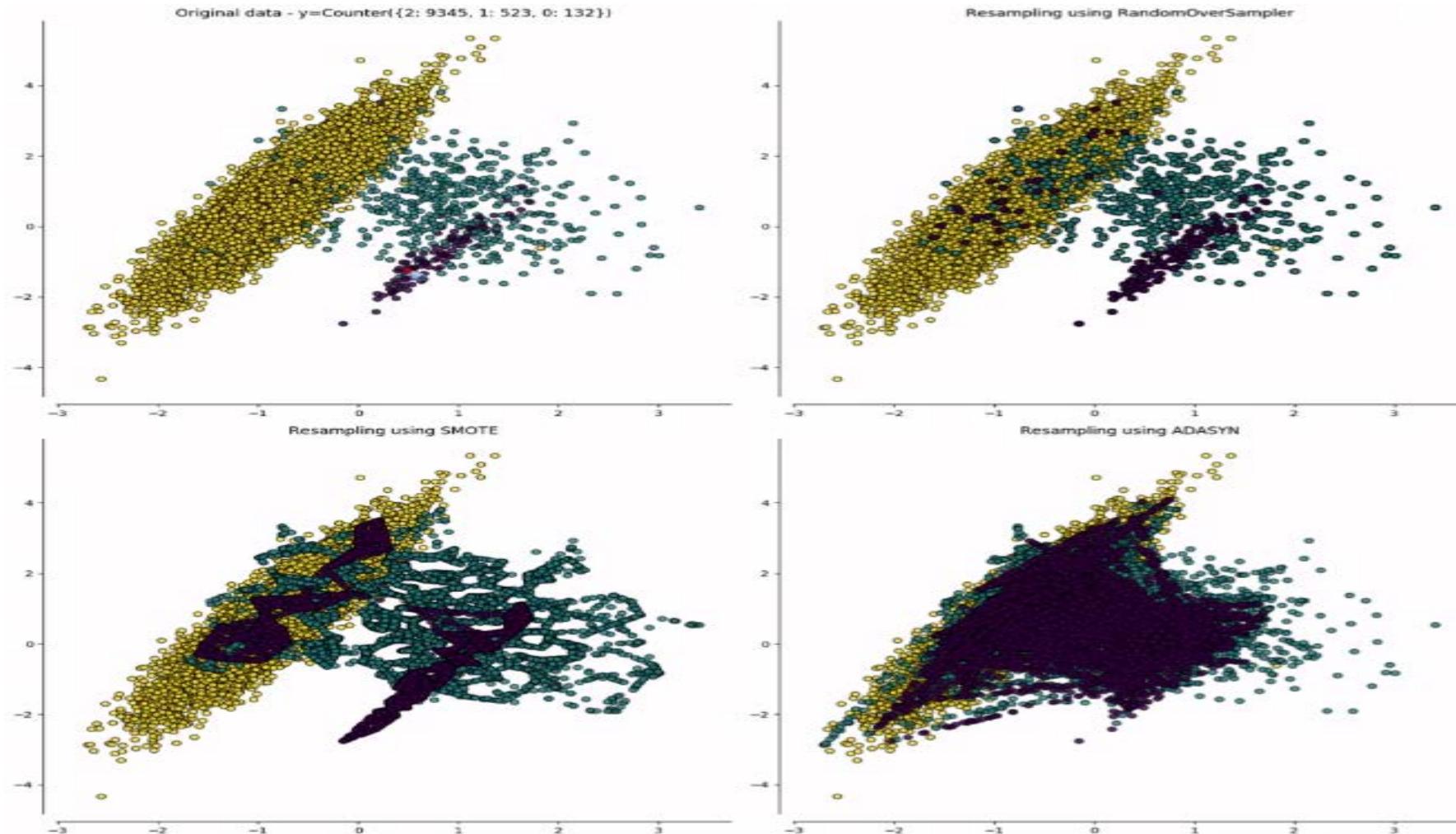
Unit 02 | Imbalanced Dataset

Over Sampling

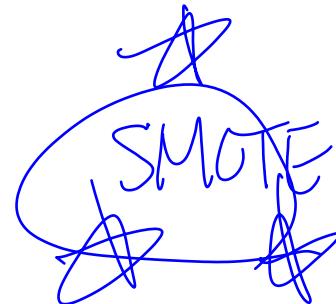
2. SMOTE – 실제 데이터의 거리 사이에 데이터를 만들어내자



Unit 02 | Imbalanced Dataset



Unit 01 | Feature Engineering



보통 가장 좋은방법!!

Sampling 실습 켜주세요!

over or under
sampling이
그냥 좋은방법은 아님.

정보의 양이 너무 많아도 안좋다
under sampling 하면 좋지 않음!

Unit 01 | Feature Engineering

Unit 02 | Imbalanced Dataset

Unit 03 | Cloud Service

Unit 03 | Cloud Service

오늘은 머신러닝 마지막 시간

많이 하는 질문 :

- 딥러닝 하면 노트북으로 버겁지 않을까요?
- 데이터 커지면 돌리는데 너무 힘들어요 ...
- 앞으로 프로젝트를 해야 하는데 컴퓨팅 파워가 너무 구려요



안녕하세요! 투빅스에
관심있는 학생입니다
노트북이 i3인데
괜찮을까요?

그래서 준비하였습니다!

Unit 03 | Cloud Service

Google Colaboratory

구글 Colaboratory(이하 Colab)은 구글 드라이브에 연결하여 사용할 수 있는 주피터 노트북 호환 서비스이다. 2017년 10월에 공개된 Colab은 별다른 설치 없이 웹 브라우저 만을 이용해 주피터 노트북과 같은 작업을 할 수 있고 다른 사용자들과 공유가 쉬워 연구 및 교육용으로 많이 사용되고 있다



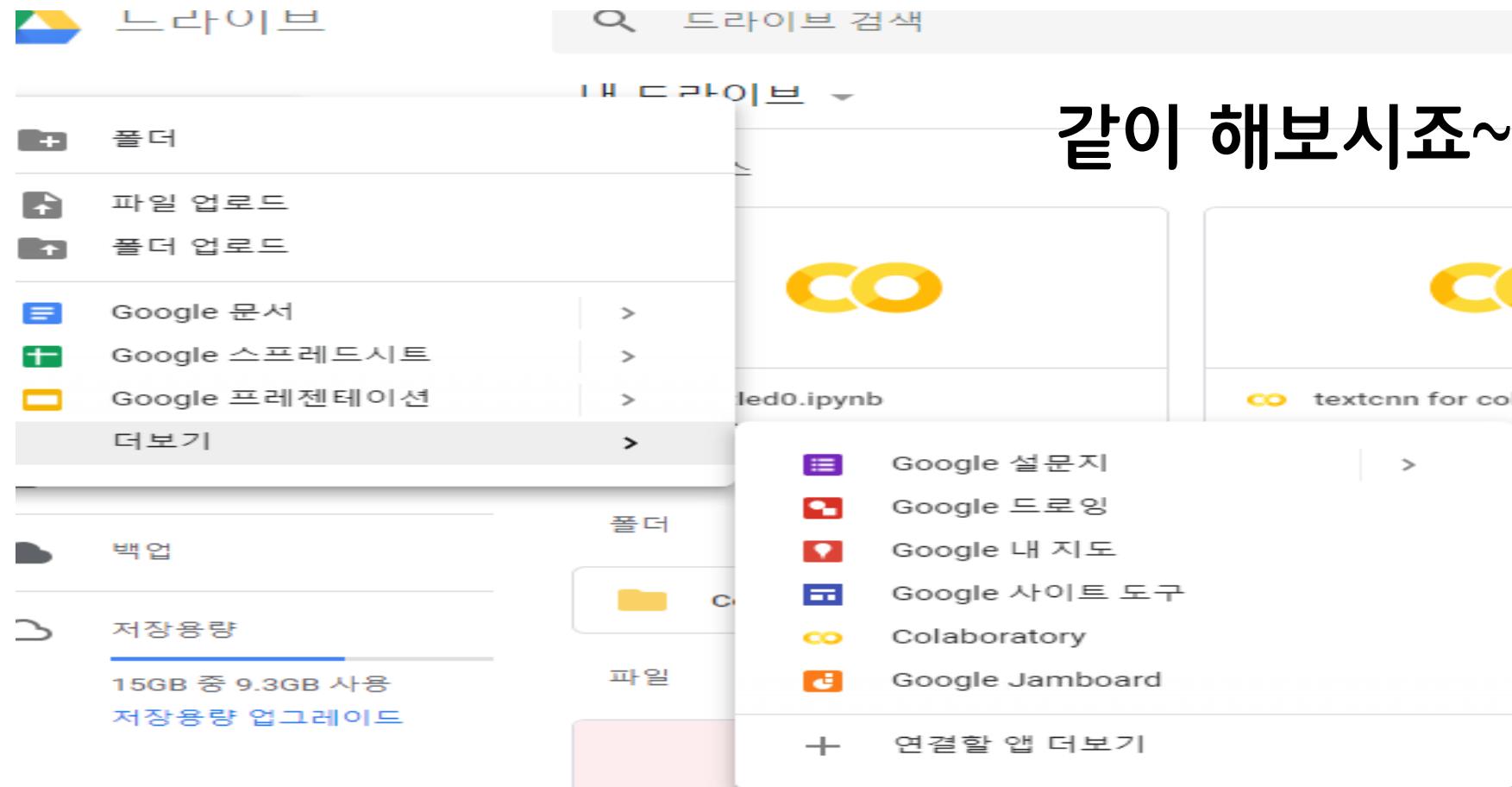
Unit 03 | Cloud Service

Google Colaboratory

1. 무료로 GPU를 사용할 수 있다.
2. Tensorflow, keras, matplotlib, scikit-learn, pandas 등 데이터 분석에 많이 사용되는 패키지들이 미리 설치되어 있다.
3. 코드 셀 내에서 필요한 패키지를 설치하고 환경설정을 할 수 있다. 이 점은 서로다른 환경을 가진 다른 사람과 협업을 할 때 환경을 동일하게 구성해야 하는 수고를 덜어 준다.
4. 구글 드라이브나 구글 스프레드시트 등과 같은식으로 공유와 편집이 가능하다. 만약 두 명 이상의 사람이 동시에 같은 파일을 수정 하더라도 변경사항이 모든 사람에게 즉시 표시된다.

Unit 03 | Cloud Service

Google Colaboratory



Unit 03 | Cloud Service

Google Colaboratory

The screenshot shows the Google Colaboratory interface. On the left, there is a code editor with several cells containing Python code. Cell [1] imports `sklearn.datasets`. Cell [3] loads a wine dataset and prints its shape. Cell [7] imports `sklearn.linear_model` and `cross_val_score`. Cell [8] shows the result of a cross-validation score. A context menu is open over cell [1], with the option "런타임 유형 변경" highlighted.

On the right, a "노트 설정" (Notebook Settings) dialog is open. It shows the current runtime type as "Python 3". Under "하드웨어 가속기" (Hardware Accelerator), "GPU" is selected. There is a checkbox for "이 노트를 저장할" (Save this notebook) which is unchecked. Below the dialog, there are buttons for "None", "GPU", "TPU", "최소" (Minimum), and "저장" (Save).

Unit 03 | Cloud Service

Data Loading 방법

https://drive.google.com/drive/folders/1IPYb_F6vjsJiD600z4eaE_yX5BetYhfE?usp=sharing

Assignment

과제 : Colab으로 과제를 하고 구글 클라우드에 올리기.

Assignment : Colab으로 제공된 wine2.csv데이터를 통해 Imbalanced Problem을 오늘 배운 Random, SOMTE OverSampling으로 해결하고 배운 분류 모델을 3개 이상 활용하여 Oversampling 하기 전 성능과 비교해주세요.(성능이 떨어질 수 있음.)

https://drive.google.com/drive/folders/1IPYb_F6vjsJiD600z4eaE_yX5BetYhfE?usp=sharing

FINAL MACHINE LEANRING



Q & A

들어주셔서 감사합니다.