DACON 스터디 2주차

End-to-End Machine Learning Project

이제윤

CONTENTS

End-to-End ML project

[한 줄의 코드로 자동학습! 머신러닝을 자동화하는 AutoML]

[Auto ML의 예 PyCaret]

Using AutoML is simple and even POWERFUL.

fit API

지금 train.csv라는 테이블 데이터를 갖고 있고, 예측하고 싶은 라벨이 'class'라는 열에 들어 있다고 합시다. 이때 AutoGluon-Tabler에서는 다음과 같은 세 줄의 Python 코드만으로 학습과 예측이 가능합니다.

```
from autogluon import TabularPrediction as task #AutoGluon의 로딩
predictor = task.fit("train.csv", label="class") #학습
predictions = predictor.predict("test.csv") #테스트 데이터에 대한 예측
```

이것만으로도 태스크가 분류인지 회귀인지, 분류라면 어떤 클래스의 분류인지를 자동으로 판정하고 다양한 모델을 앙상블 시켜 강력한 모델을 학습시킵니다. 이처럼 AutoGluon-Tabuler는 심플함을 달성한 누구나 사용하기 쉬운 프레임워크라고 할 수 있습니다.

한 줄의 코드로 자동학습! 머신러닝을 자동화하는 AutoML

1 Data Preprocessing

보다 높은 정확성을 갖는 분석을 위해 원자료에 대해 전환 및 가공을 거치는 단계
→ AutoML의 등장으로 그 중요도 및 비중이 커지고 있음

정규화와 표준화

특성변수의 단위 등에서 나타나는 차이를 조정해주는 역할

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

$$z = \frac{x - \mu}{\sigma}$$

차원축소: PCA / t-sne

feature가 너무 많으면 오버피팅 가능성이 있기 때문에 차원축소 필요

One-hot encoding

범주형 변수를 수치형 변수로 변환

```
[[0. 0. 1. 0. 0. 0. 0. 0.] # 인덱스 2의 원-핫 벡터
 [0. 0. 0. 0. 0. 1. 0. 0.] # 인덱스 5의 원-핫 벡터
 [0. 1. 0. 0. 0. 0. 0. 0.] # 인덱스 1의 원-핫 벡터
 [0. 0. 0. 0. 0. 0. 1. 0.] # 인덱스 6의 원-핫 벡터
 [0. 0. 0. 1. 0. 0. 0. 0.] # 인덱스 3의 원-핫 벡터
 [0. 0. 0. 0. 0. 0. 0. 1.]] # 인덱스 7의 원-핫 벡터
```

이상치 및 결측치 처리

머신러닝 모형은 직접 결측치를 처리할 수 없음

bag of words

텍스트를 수치형 변수로 변환해주는 방법(NLP)

the dog is on the table

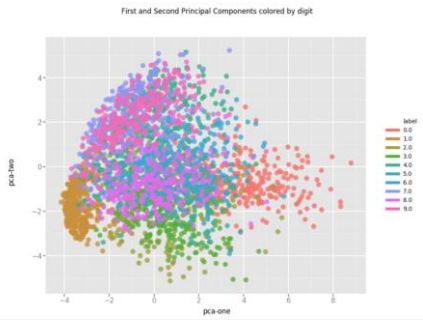
are	cat	dog	is	now	on	table	the
0	0	1	1	0	1	1	1

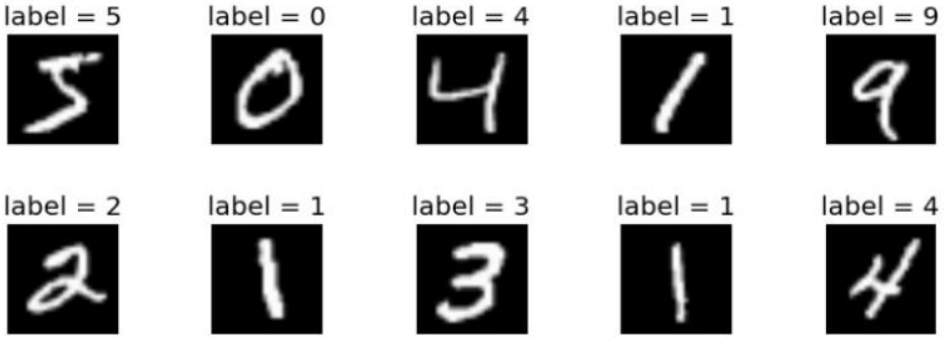
불균형 자료 처리: SMOTE / ADASYN

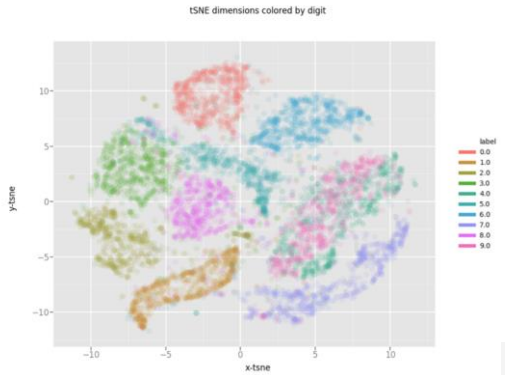
불균형 자료 문제를 해소하기 위한 과대표집 방법

1 Data Preprocessing : 차원 축소

PCA vs t-SNE







PCA	t-sne
Principal Component Analysis	t-distributed stochastic neighbourhood embedding
linear dimensionality reduction technique	non-linear Dimensionality reduction technique
reduce the dimensionality of data that is highly correlated by transforming the original set of vectors to a new set	minimize the Kullback–Leibler divergence (KL divergence) between the two distributions

t-SNE 개념과 사용법 - gaussian37

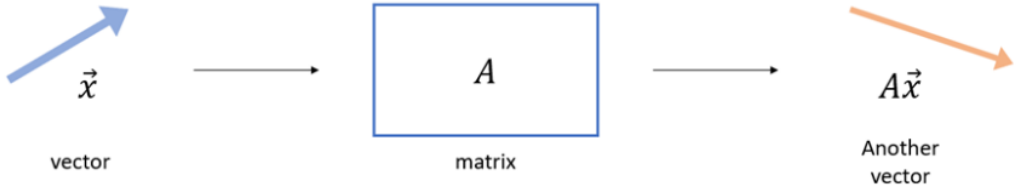
https://huidea.tistory.com/44

https://hong-sik.tistory.com/22

1 Data Preprocessing : 차원 축소 PCA

벡터와 행렬 연산에 대하여

벡터에 행렬 연산을 취해준다는 것은 벡터를 변환시켜 다른 벡터를 출력해주는 것을 의미한다.
변환 후의 벡터는 변환 전의 벡터와 비교했을 때, 크기도 방향도 모두 변할 수 있다.



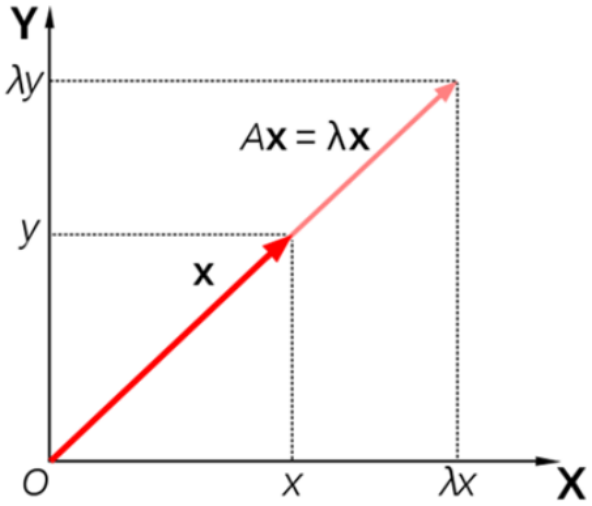
행렬의 고유값과 고유벡터

그런데 특정한 벡터와 행렬은 변환을 취했을 때, 방향은 바뀌지 않고 크기만 바뀔 수 있다.
이 말은 즉, 어떠한 벡터에 행렬 A 연산을 취하여 변환시킨 값이 상수배라는 뜻이다.

임의의 n×n 행렬A에 취했을 때 크기만 바뀌게 하는 0이 아닌 벡터가 존재한다면
숫자 람다는 행렬 A의 고유값이라고 할 수 있고 이 때 벡터는 고유값 람다에 대응하는 고유벡터이다.

$$A\vec{x} = \lambda\vec{x}$$ → 벡터

임의의 n X n 행렬 / 상수, 고유값



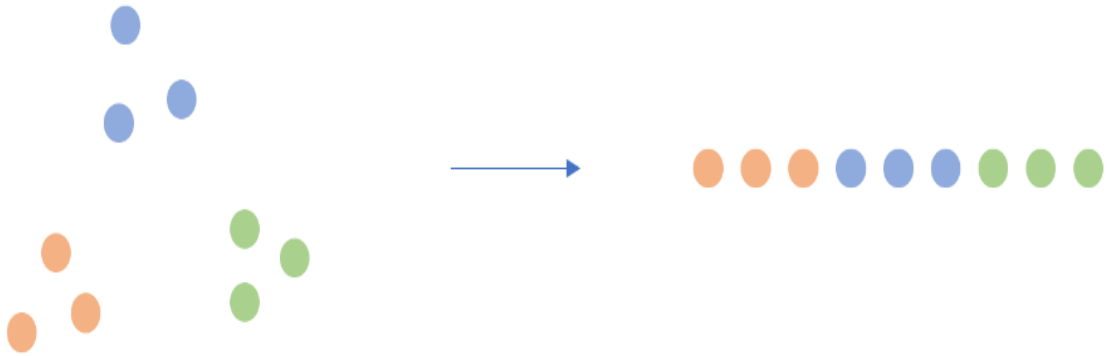
람다값(고유값)이 크면 선이 더 멀리 뻗어나가고 이는 분산이 크다는 것을 의미한다.

그렇게 구해진 고유벡터1을 PC1이라하고 고유벡터2를 PC2라고 한다.
PC는 차원의 개수만큼 구할 수 있다.

공분산 행렬의 고유값과 고유벡터

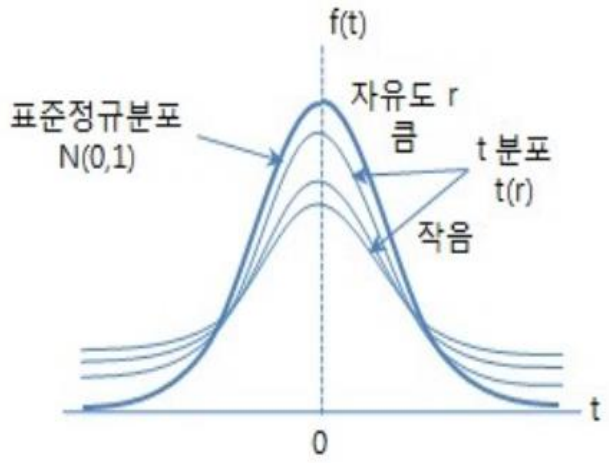
공분산행렬의 고유벡터는 데이터가 어떤 방향으로 분산되었는지를 나타낸다.
고유값은 고유벡터 방향으로 얼마만큼의 크기로 벡터 공간이 늘려지는지를 나타낸다.
고유값이 큰 순서대로 고유벡터를 정렬하는 것은 중요한 순서대로 주성분을 구하는 것을 의미한다.

1 Data Preprocessing : 차원 축소 t_sne



SNE 학습과정에 사용되는 가우시안 분포는 t 분포에 비해, 거리에 따른 확률 값 변화의 경사가 가파른 특징을 갖는다. 따라서 특정 거리 이상부터는 학습과정에 거의 반영이 되지 않는 문제점을 가지고 있으며, 이를 Crowding Problem 이라고 한다. 이러한 문제점을 해결, 보완하기 위해 고안된 방법이 t-SNE이다.

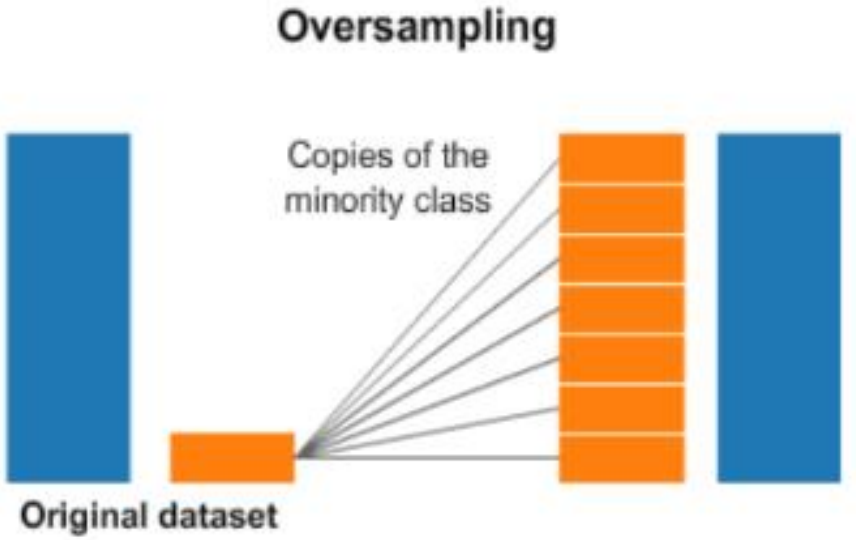
쉽게 말해, **SNE는 기존에 이웃이었던 점들을 차원축소 후에도 이웃으로 유지시키는 로직**이다. 위 그림을 보면 2차원에서 파란색, 주황색, 초록색 점들은 각각 서로 가까이 모여 있었고, 차원축소 후에도 서로 모여있음을 확인할 수 있다. 하지만 파란색-주황색, 파란색-초록색, 주황색-초록색은 2차원에서 어느 정도 떨어져 있었음에도 불구하고 1차원에서는 가까워져 있다. 이렇게 기존 차원에서 멀리 떨어져 있었던 점들이 SNE에 의해서는 가까워질 수 있다.



[그림 6. t 분포가 가우시안 분포에 비해 완만함을 확인 할 수 있음]

t-SNE 는 t-distribution을 사용함으로써 군집 간의 거리를 벌릴 수 있다.

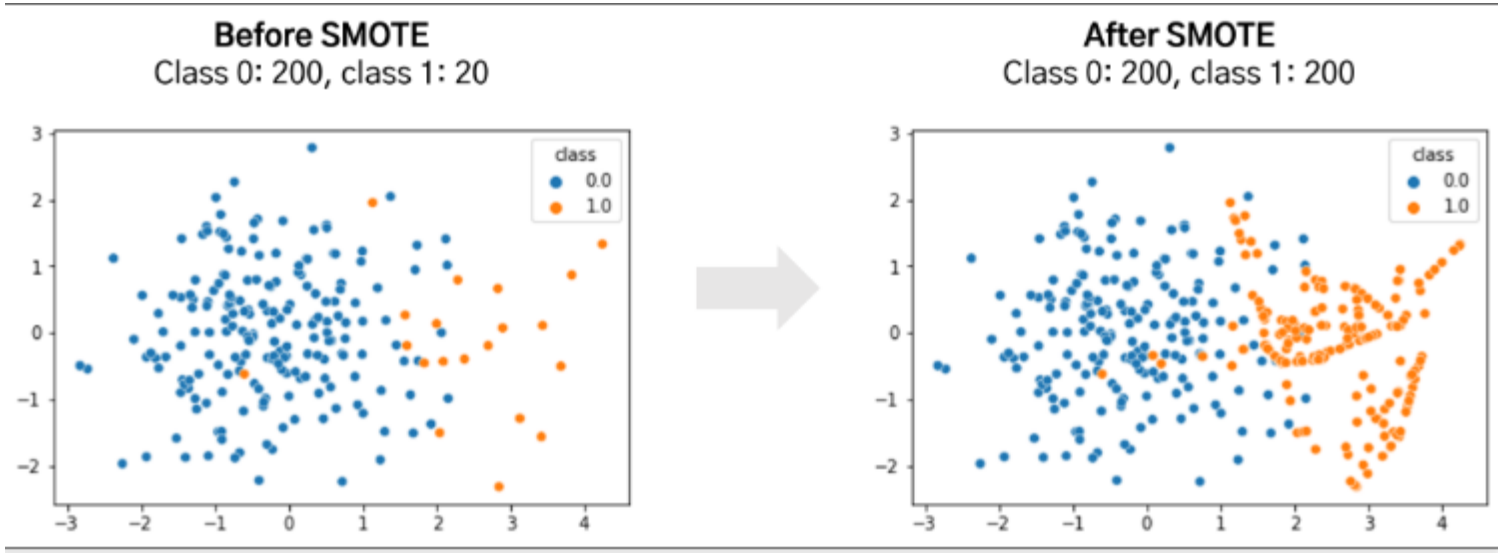
1 Data Preprocessing : 불균형 자료 -> oversampling



SMOTE	ADASYN
Oversampling	Oversampling
K- nearest neighbors	SMOTE의 개선된 버전
$x_{syn} = x_i + \lambda\ (x_k - x_i),\ x_k \in S_i$	표본 수에 대한 weight를 통해 추출

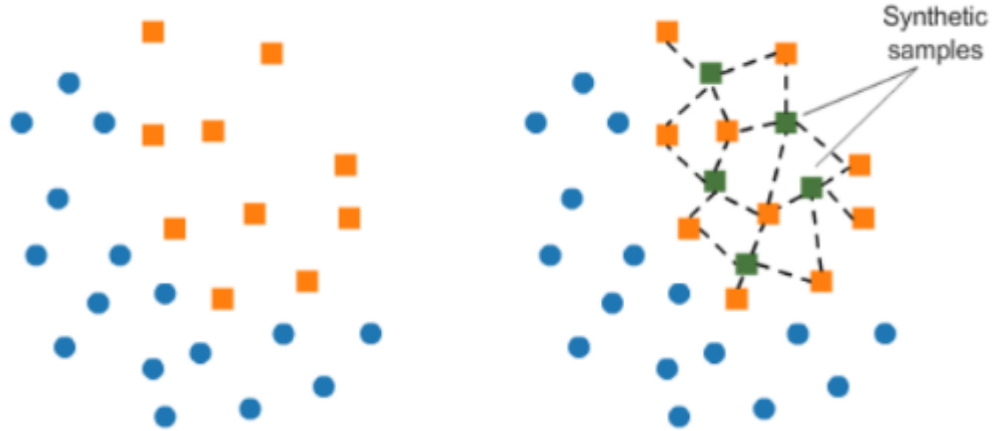
1 Data Preprocessing : oversampling : Smote

SMOTE - 인코딥, 생물정보 전문위키

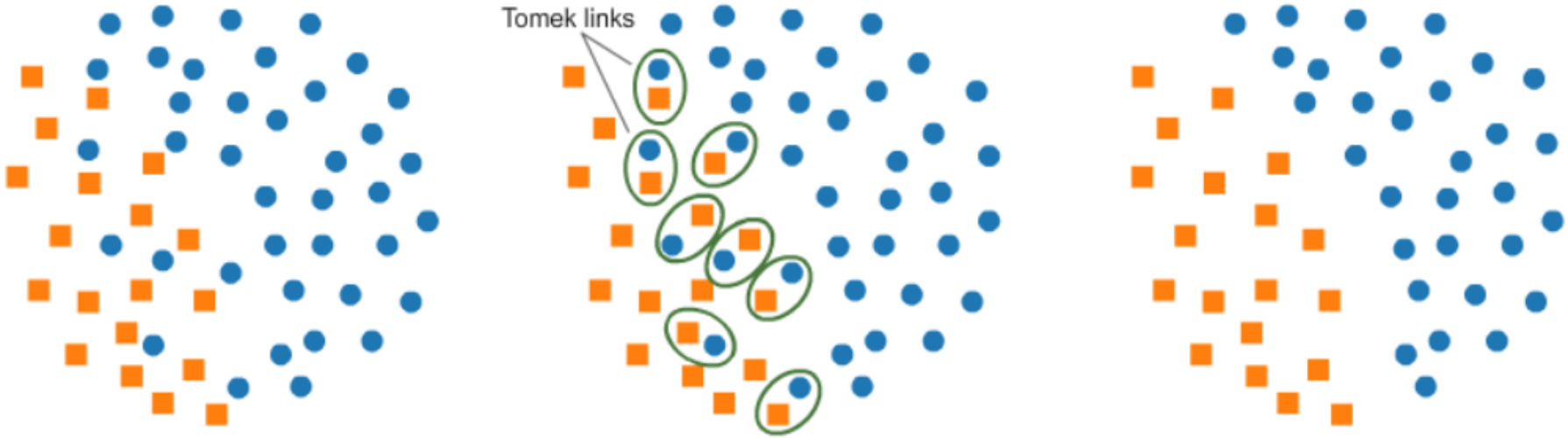


SMOTE: Synthetic Minority Over-sampling Technique

말 그대로 소수 군집의 데이터를 “고르게(synthetic)” 생성함.



1 Data Preprocessing : oversampling : ADASYN



ADASYN : Adaptive Synthetic Sampling Approach

2 Learning Process

learning	목적	구분
K-nearest neighbors (KNN)	분류, 회귀(4장)	지도학습, 사례기반, 배치
Kernel smoothing	density estimation(4장)	
Adaptive linear neuron	분류(5장)	지도학습, 모형기반, 배치
Logistic regression	분류(5장)	지도학습, 모형기반, 배치, online
Discriminant analysis (안쓸듯)	분류(6장)	지도학습, 모형기반, 배치
Naive Bayes	분류(6장)	지도학습, 모형기반, 배치
Classification and Regression Tree (CART)	분류, 회귀(7장)	지도학습, 배치, 비모수
Support vector machine (SVM)	분류(8장), 회귀(11장)	지도학습, 모형기반, 배치, online
Kernelized SVM (kernel trick)	비선형분류(8장), 비선형회귀(11장)	지도학습, 모형기반, 배치, online
Principal component analysis (PCA)	차원축소(9장)	비지도학습, 모형기반, 배치
Kernelized PCA	비선형 차원축소(9장)	비지도학습, 모형기반, 배치
Linear discriminant analysis (LDA), MDS, Manifold Learning, t-SNE	차원축소(9장)	비지도학습, 모형기반, 배치
Regression (OLS)	회귀(11장)	지도학습, 모형기반, 배치, online
RANSAC → outlier의 민감하지 않은 회귀방법	로버스트 회귀(11장)	지도학습, 모형기반, 배치

learning	목적	구분
Bagging	분류, Ensemble(12장) = bootstrap	지도학습, 모형기반, 배치
Boosting, Random forest	분류, 회귀, Ensemble(12장)	지도학습, 모형기반, 배치
Xgboost, LightGBM, Catboost	분류, 회귀, Ensemble(12장)	지도학습, 모형기반, 배치
K-means clustering	군집(13장)	비지도학습, 사례기반, 배치
Hierarchical clustering	군집(13장)	비지도학습, 사례기반, 배치
DBSCAN, HDBSCAN	군집(13장)	비지도학습, 사례기반, 배치
Sentiment analysis	분류, 회귀, 문서분석(14장)	지도학습, 모형기반, 배치, online
Multilayer Neural Network/backpropagation	딥러닝의 기초이론	지도학습, 모형기반, 온라인
Convolutional Neural Network	비정형데이터(이미지, 텍스트, 오디오, 음성)	지도학습, 모형기반,온라인
Recurrent Neural Network/LSTM	자연어 처리(언어번역, 감성분석, 고객서비스 자동화, 웹 검색)	지도학습, 모형기반, 온라인

2 Learning Process

Train / Test / Validation set의 차이 :: 프라이데이

Machine Learning Model Evaluation Metrics | Kaggle

<Validation set & Test set>

validation set은 학습이 이미 **완료된** 모델을 **검증**하기위한 dataset이다.

test set은 **학습과 검증이 완료된** 모델의 성능을 평가하기위한 dataset이다.

보통 Train : Test 데이터를 8 : 2로 나누는데 여기서 Train 데이터 중 일부를 validation set으로 이용해

결국 Train : Validation : Test 을 일반적으로 6 : 2 : 2로 이용한다.

학습데이터를 이용해 학습된 모형을 검증데이터에 적합시켜 학습데이터에서 구현된 성능과 비슷한 수준인지 점검하고 초모수 조정

3 Model Evaluation

Contents

Gridsearch

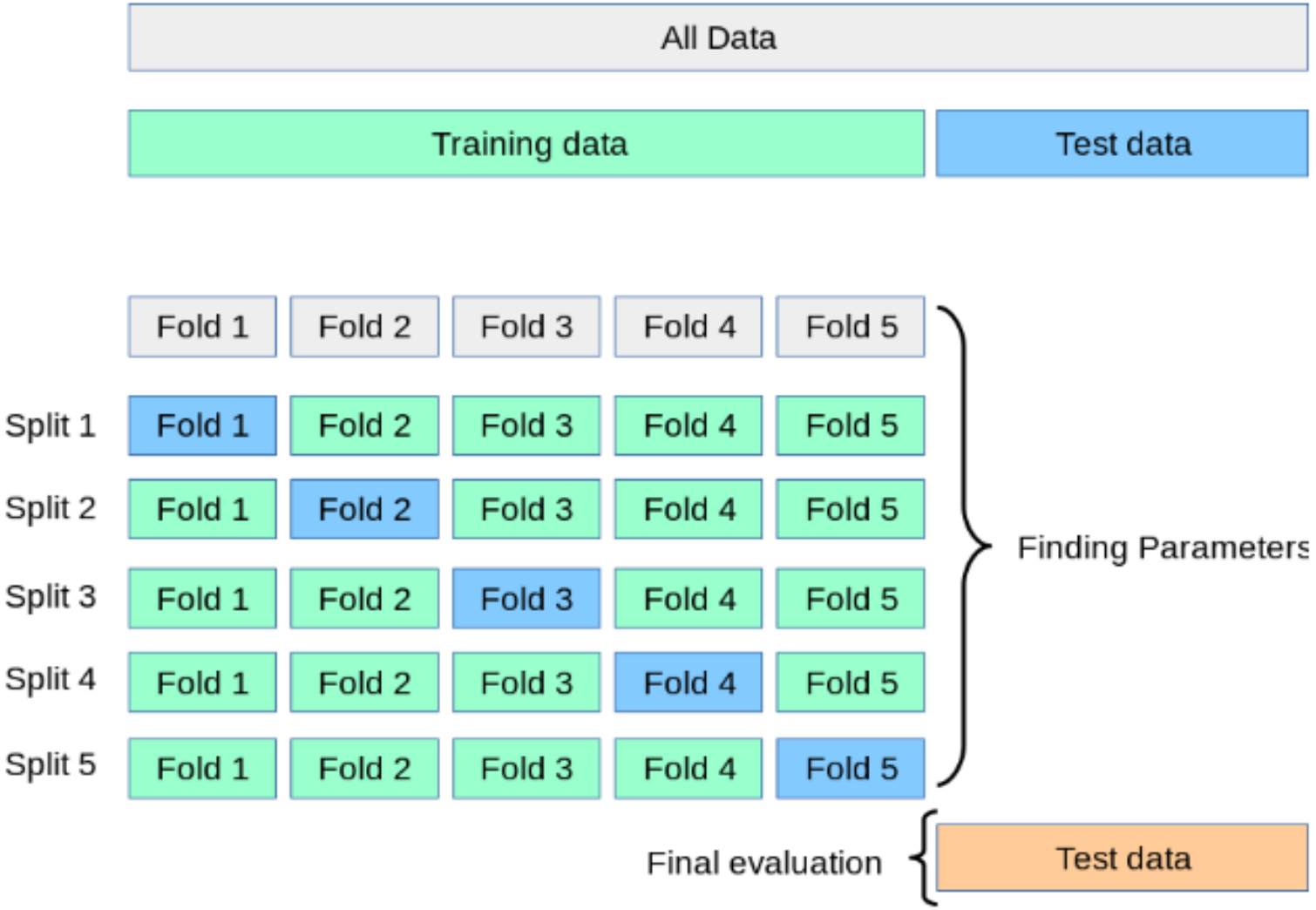
```
param_grid = {
    'n_estimators': [100, 150, 200, 250],
    'max_depth': [None, 6, 9, 12],
    'min_samples_split': [0.01, 0.05, 0.1],
    'max_features': ['auto', 'sqrt'],
}
```

GridSearch?

GridSearch 는 우리가 지정해준 몇 가지 **잠재적 Parameter들의 후보군들의 조합 중에서 가장 Best 조합을 찾아줍니다**. 어떻게 보면 우리가 하나하나 대입해 가면서 loss를 확인하는 작업을 GridSearch는 대신 해준다고 보면 됩니다. 또한, sklearn 패키지에서 제공해주고 있기때문에 매우 손쉽게 사용할 수 있습니다.

하지만, 가장 큰 단점은 우리가 지정해 준 hyperparameter 후보군의 갯수만큼 비례하여 시간이 늘어기 때문에 **최적의 조합을 찾을 때까지 시간이 매우 오래 걸린다는 단점**이 있습니다.

CV (cross validation)



GridsearchCV = GridSearch + CrossValidation

```
model=xgb.XGBClassifier()
param_grid={'booster' :['gbtree'],
                  'silent':[True],
                  'max_depth':[5,6,8],
                  'min_child_weight':[1,3,5],
                  'gamma':[0,1,2,3],
                  'nthread':[4],
                  'colsample_bytree':[0.5,0.8],
                  'colsample_bylevel':[0.9],
                  'n_estimators':[50],
                  'objective':['binary:logistic'],
                  'random_state':[2]}
```

```
# 3번
cv=KFold(n_splits=6, random_state=1)

# 4번
gcv=GridSearchCV(model, param_grid=param_grid, cv=cv, scoring='f1', n_jobs=4)
```

Gridsearch 와 CV 모두 모든 경우의 수를 따져보는 것이고

둘을 같이 사용하는 경우가 많기 때문에

Scikit-learn에서 두 과정을 한 번에 진행할 수 있는 메서드를 제공함.

https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=gustn3964&logNo=221431933811

4 Prediction

Reference

How to Develop an End-to-End Machine Learning Project and Deploy it to Heroku with Flask

파이썬 원-핫 인코딩(One-hot encoding) 정리 - GROWTH.J

데이터 표준화에 대한 질문입니다. - 인프런 | 질문 & 답변

파이썬 원-핫 인코딩(One-hot encoding) 정리 - GROWTH.J

오버샘플링 기법(Over Sampling Methods)

주성분 분석(PCA)를 이해해보자

t-SNE

차원축소, 시각화 도구: t - SNE (Stochastic Neighbor Embedding) : 네이버 블로그

Train / Test / Validation set의 차이 :: 프라이데이

⚖️ Machine Learning Model Evaluation Metrics | Kaggle

HandsonMachineLearning 2장 End-to-End ML project.

KUBIG 2022-SPRING ML STUDY.

Hands on machine Learning 실습 (20분 정도)

End-to-End ML ipynb Google Colab link

개인 과제 (월요일 자정까지)

타이타닉 생존 예측 경진대회 – DACON

데이터 불러오기, 간단한 전처리, 모델링, 결과 파일 생성해서

1. DACON에 제출한 결과 리더보드 스크린샷 올리기

2. google colab 다운 받고 week2_hw_jeyun.ipynb 형식의 파일로 코드 작성해서 공유 드라이브에 업로드하기.

week2 과제를 열심히 할수록 다음주 분류 모델링에서 좋은 성능을 낼 수 있음.
Dacon에 공유된 코드 있으니까 얼마든지 따라해도 좋음
대신 복붙만 하지 말고 코드를 이해하고 외워보려고 노력하기