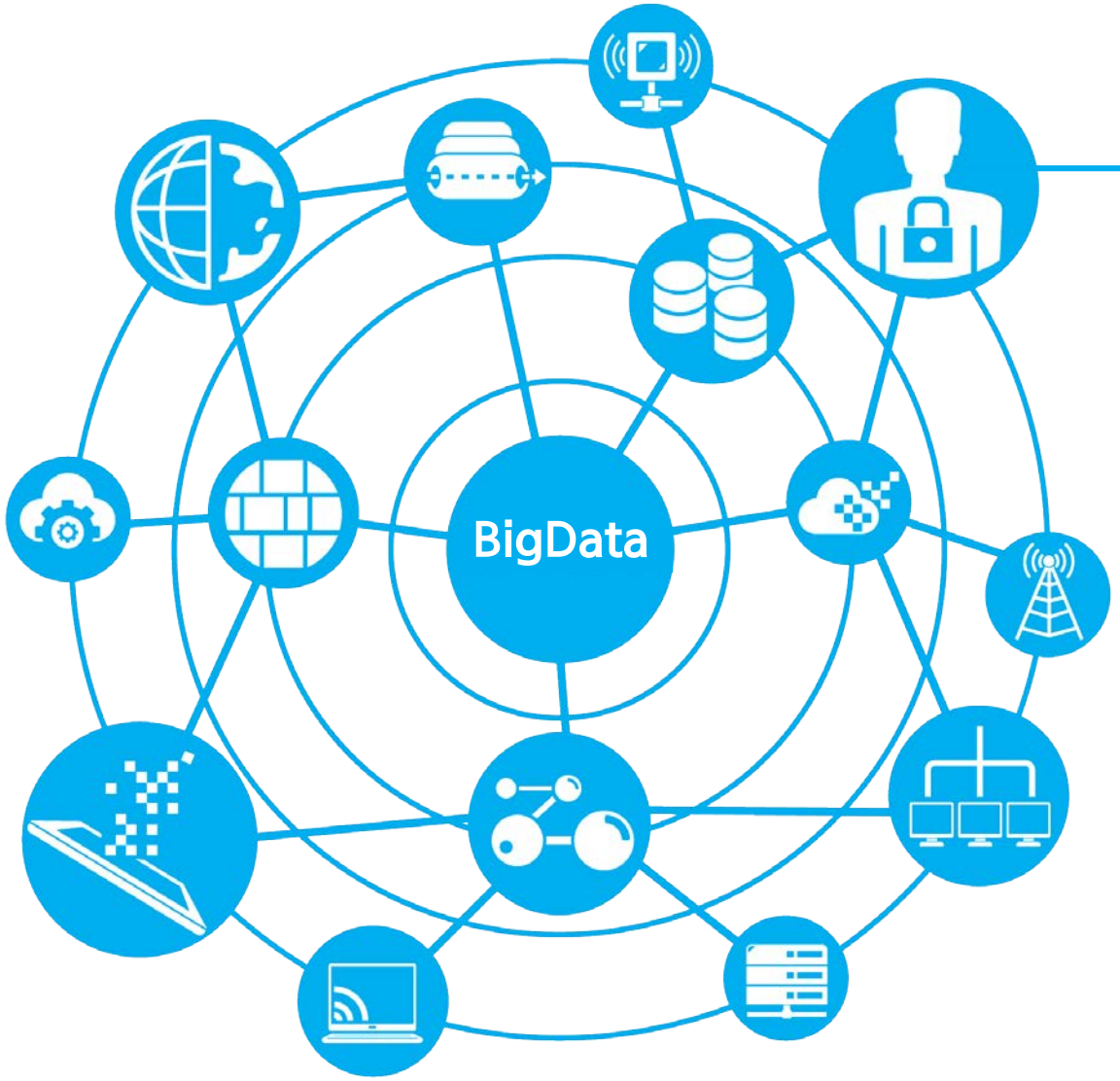


ISHY x Bigcontest

기계학습 기반 보험 사기자 예측

사용한 알고리즘 : Random forest

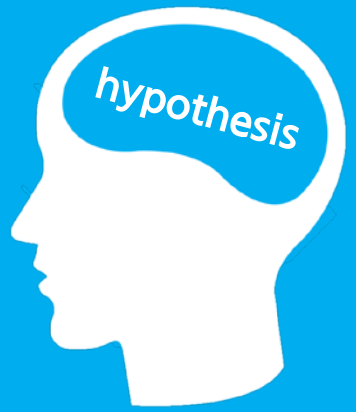
사용한 언어 : Python

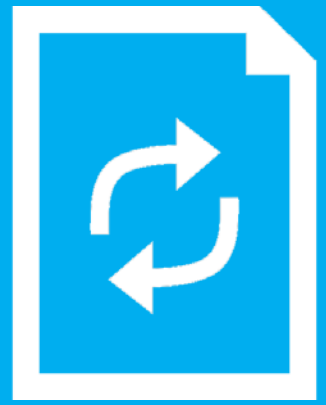
작업환경 : Jupyter

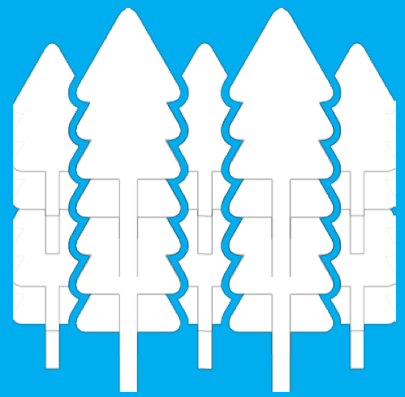
ISHY x Bigcontest challenge league

Total process

1. Original Data
2. Hypothesis
3. Preprocessing
4. Random Forest







Original Data

Hypothesis

Preprocessing

Random Forest

Original data

BGCON Data

CLAIM_DATA
109020 x 39
보험청구 데이터

CUST_DATA
22400 x 25
고객 데이터

CNTT_DATA
113010 x 21
보험계약 데이터

FPINFO_DATA
31522 x 7
FP 데이터

FMLY_DATA
426 x 3
고객가족 데이터

Original data

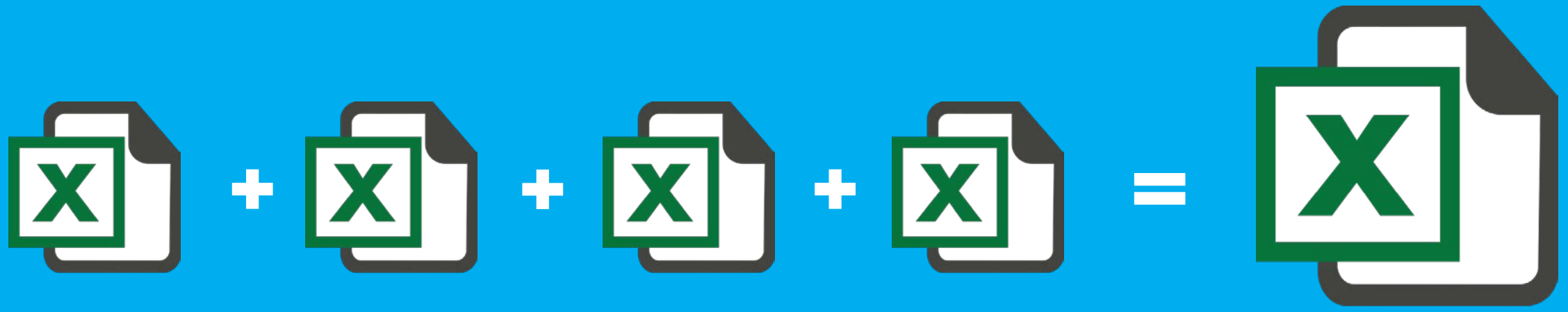
BGCON Data

대표적인 비 정형 데이터이다. 수치형, 범주형 데이터들이 섞여있고, 빈 칸, 오타 등도 존재하여, 가장 먼저 Data cleaning부터 시행했다. 다음 단계로는 여러 알고리듬 중에서 이러한 비정형 데이터를 분석하는데 적합한 알고리듬을 고르기 위해 고민을 하였다.

CLAIM_DATA	CUST_DATA	CNTT_DATA	FPINFO_DATA	FMLY_DATA
109020 x 39	22400 x 25	113010 x 21	31522 x 7	426 x 3
보험청구 데이터	고객 데이터	보험계약 데이터	FP 데이터	고객가족 데이터

MERG_DATA

단 하나의 data set을 만들기 위해 사용하기 어려운 FMLY_DATA는 과감히 버리고,
나머지 4개의 DATA를 합쳐서 보험청구 기반으로 통합된 MERG_DATA를 만들었다.



CLAIM_DATA
109020 x 39
보험청구 데이터

CUST_DATA
22400 x 25
고객 데이터

CNTT_DATA
113010 x 21
보험계약 데이터

FPINFO_DATA
31522 x 7
FP 데이터

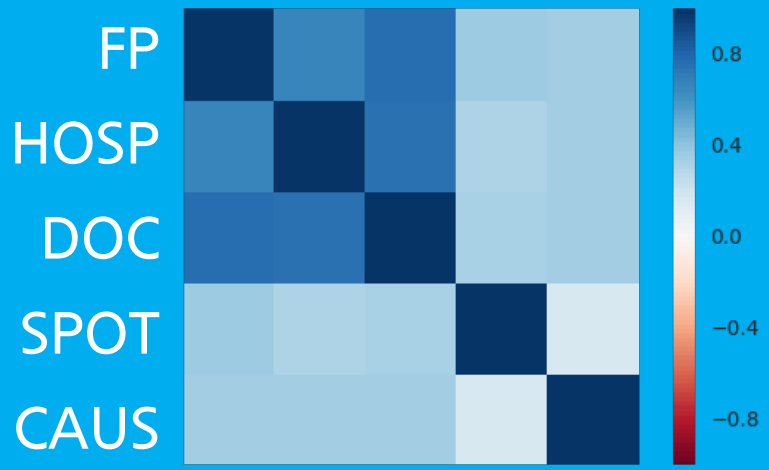
MERG_DATA
109020 x 88
보험청구 기준 통합 데이터

Hypothesis

우리의 속담 "세 살 버릇, 여든까지 간다." 와 "끼리끼리 논다." 에서 영감을 얻었다.
우리는 사기를 주로 치는 누군가가 대부분 사기를 칠 것이고,
그런 사람들끼리 서로 연관관계가 있을 것으로 생각했다.

FP, 의사, 각 지점 등에 대해 유의값을 구하였고
특히 FP, 병원, 의사 사이의 상관관계가 높은 것을 확인하였다.

ex) FP A의 유의값 : FP_PROB = $\frac{\text{A의 보험사기 건수}}{\text{A의 전체지급 건수}}$



Preprocessing

기존 88개 중 선정한 35개 항목
&
새로 만든 12개 항목

총 47개 항목

수치형 36 항목 / 범주형 11 항목

기존 항목 중 선정 기준

- ✓ 다른 항목과 중복 정도 — ex) 중복되는 항목
- ✓ 범주형 데이터의 복잡성 — ex) 병명
- ✓ 분석에 불필요 판단 — ex) 고객 거주 지역

새로 만든 항목 (12개 항목)

EXTN_YM	계약소멸 여부
LAPS_YM	계약실효 여부
CNTT_QURT_LEVEL	계약체결일자
RECP_QURT_LEVEL	사고접수일자
DIFF_DATE_SIZE	사고접수일자 – 계약체결일자 간 차이
DIFF_PAYM_RESN	사유일자 – 지급일자 간 차이
AMT_EVENT	사건수
PAYM_REAL0_YN	중간 일시납 변경 여부
FP_PROC	FP 유의값
DOC_PROC	의사 유의값
HOSP_PROC	병원 유의값
DIFF_CAUS_RESL	병원 및 결과 코드 일치 여부

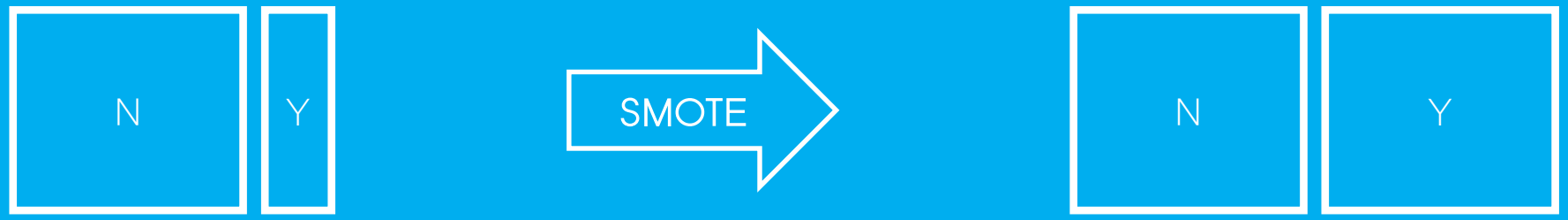
Training set revision

실제 데이터 안에는 사기자 비율이 약 22%정도로 비사기자 비율과 차이가 크다.
이런 경우 학습에서 편향이 일어날 수 있다. 따라서 이를 보정해 주어야 한다.

SMOTE (Synthetic Minority Over-sampling Technique)를 이용하여 training set 크기를 보정한다.

*SMOTE는 비율이 낮은 분류의 데이터를 임의로 만들어내는 방법이다. 분류 개수가 적은 쪽의 데이터의 샘플을 취한 뒤 k nearest neighbor을 찾고, 이웃들과의 차이를 고려하여 새로운 샘플을 만들어낸다.

우리는 2 : 7 의 데이터 비율을 SMOTE를 이용하여 1 : 1로 사기자 데이터들을 생성하였다.

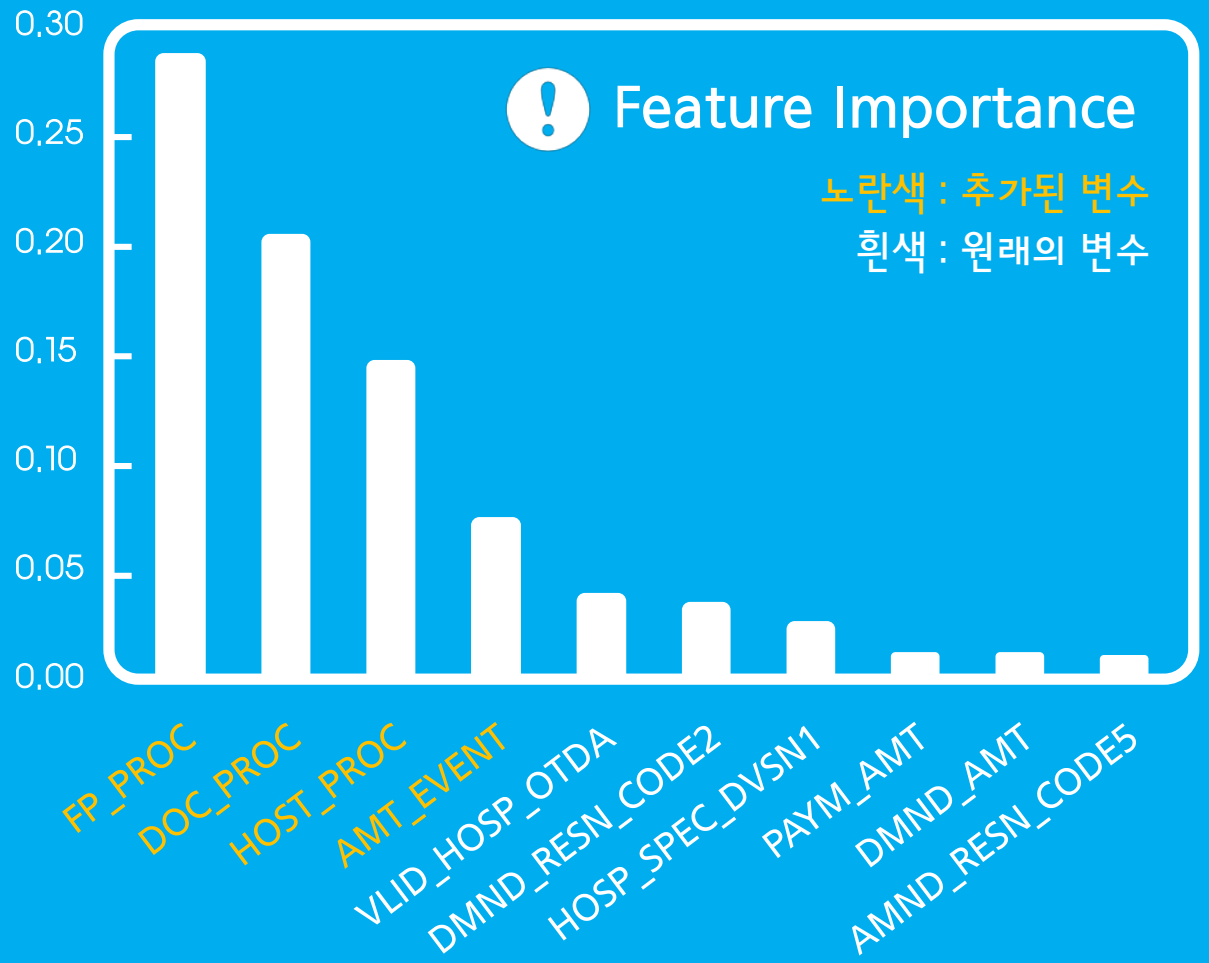


Feature Importance

Random forest를 돌릴 때 자체적으로 항목들의 중요성을 체크해준다.
총 47개의 항목 중 중요한 상위 10개를 뽑았다.

상위 10개 중 4개가 우리가 새로 만들어낸 변수
특히, FP_PROC, DOC_PROC, HOSP_PROC이 중요하게 나타났다.

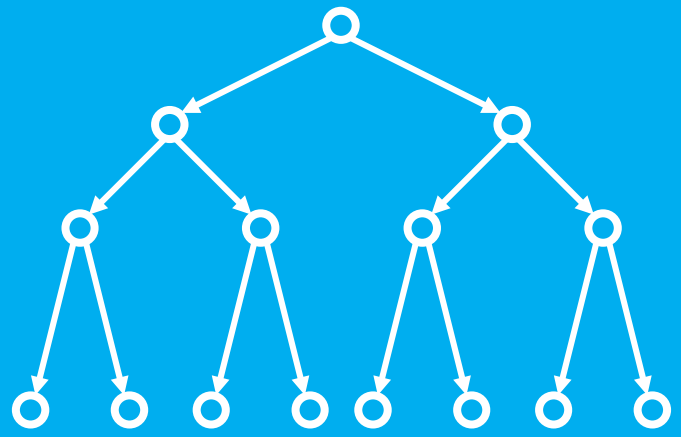
우리의 가설이 정확하게 맞아 떨어지는 것을 확인할 수 있었다.



Random forest

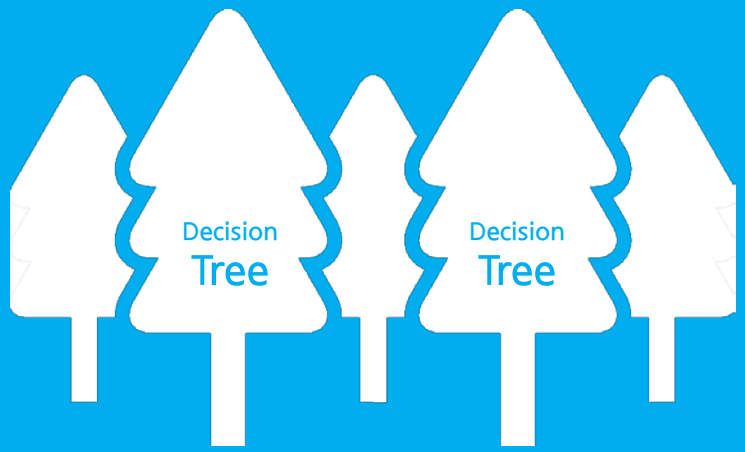
Random forest란 입력 데이터의 분류를 목적으로 회귀 분석을 하는 기계학습의 한 종류
조금씩 다른 수많은 결정 트리들을 이용하여 예측한다. 수치형 데이터는 물론 범주형 데이터도
비교적 쉽게 처리해주는 장점이 있다.

A decision tree



가 모여서

Random Forest



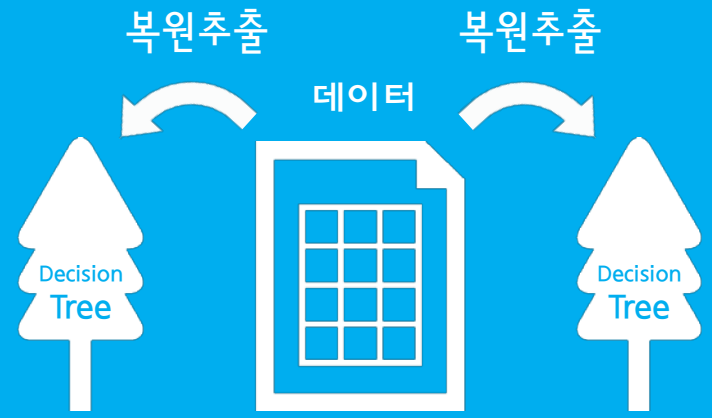
Random forest

Random forest는 다음의 두 가지 알고리듬을 사용하여 트리가 가지는 에러를 보완한다.

각 트리의 에러 = bias + variance

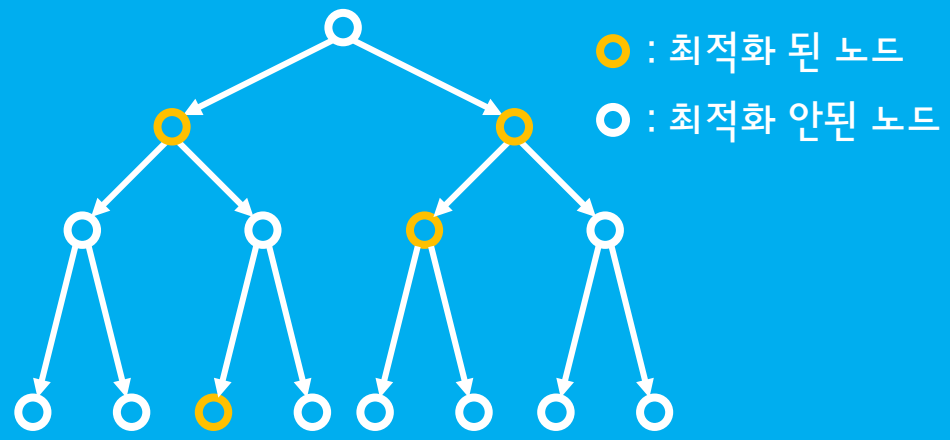
배깅 (Bootstrap aggregation; Bagging)

; 각 tree에 들어갈 train set을 같은 크기를 갖도록 복원 추출하는 것을 뜻한다.
배깅은 overfitting을 줄여준다.



임의 노드 최적화 (Randomize node optimization; RNO)

; 목적함수를 최대로 만드는 것 즉 효율적으로 분류하는 것을 최적화라고 한다. 이 때 모든 노드를 최적화 하면 overfitting이 발생할 수 있기 때문에, 특정 노드만 최적화를 시행한다. 사용할 수 있는 목적함수에는 Information gain , Gini impurity 등이 있다.우리는 Gini impurity를 사용하였다.



*Gini impurity란 각 노드에서 분류할 때 오분류가 정분류에 얼마나 섞여 있는지를 뜻한다.

QnA

저희 ISHY의 발표를 들어주셔서 감사합니다.

질문이 있으면 해주세요!

*최종적으로 주어진 데이터를 train / test set으로 약 9 : 1로 나눴을 때 자체 평과 결과는 오른쪽 표와 같이 나왔다. (트리 개수 : 1000개 경우)

**실제예측에는 유의값 대신 5단계 범주형으로 변환한 유의레벨을 사용하였다. (이유는 꽤 복잡해서 짧은 시간 내에 설명하기 힘들다…)

실제 \ 예측	N	Y
N	1880	50
Y	65	116

F measure = 0.668