2021 통계분석실습 기말 프로젝트

회사채 신용등급 변화 예측모형 개발

(2조) 1810618 이은경, 1814216 정은정, 1916129 최윤서

2021.12

프로젝트 개요

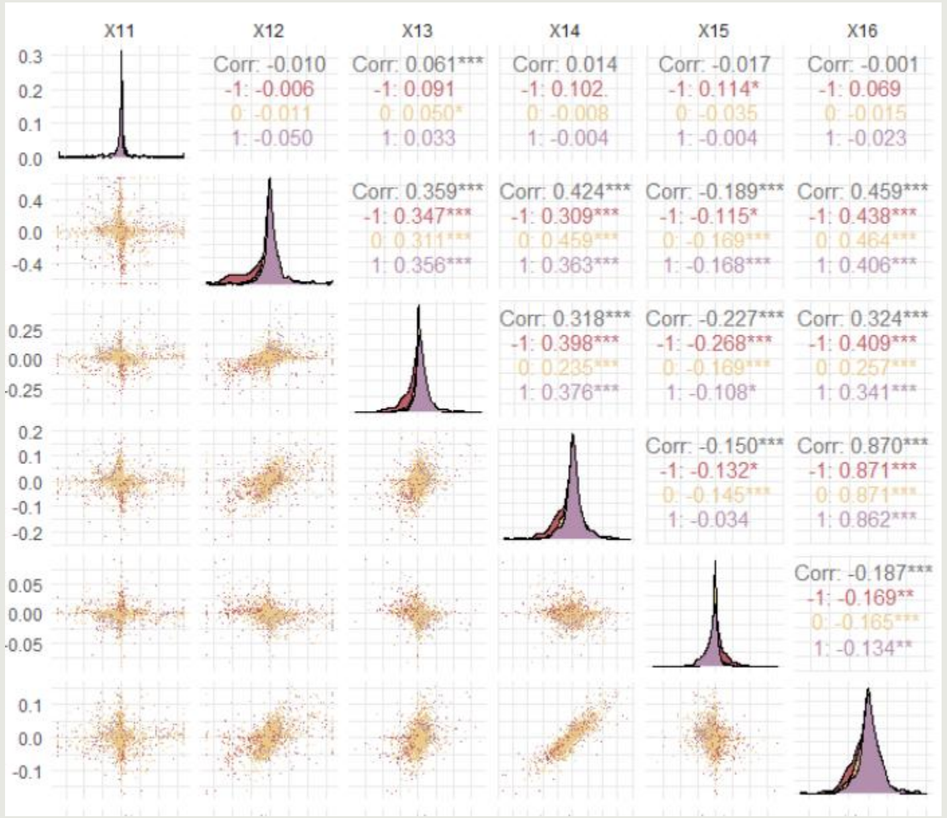
▶ 데이터 : 1998 ~ 2020까지 시간에 따라 관측된 회사 데이터

▶ 종속변수: C (신용등급 변화) (levels : +1, 0, −1)

▶ 설명변수

- yy_b : 이전 시점의 신용등급이 측정된 해
- Y_b : 이전 시점의 신용등급
- X1 ~ X5, BX1 ~ BX5 : 규모 지표
- X6 ~ X9, BX6 ~ BX9 : 비재무 지표
- X10 ~ X11, BX10 ~ BX11 : 생산성 지표
- X12 ~ X29, BX12 ~ 29 : 수익성 지표
- X30 ~ X46, BX30 ~ BX46 : 안정성 지표
- X47 ~ X56, BX47 ~ BX56 : 현금흐름 지표
- X57 ~ X60, BX57 ~ BX60 : 활동성 지표

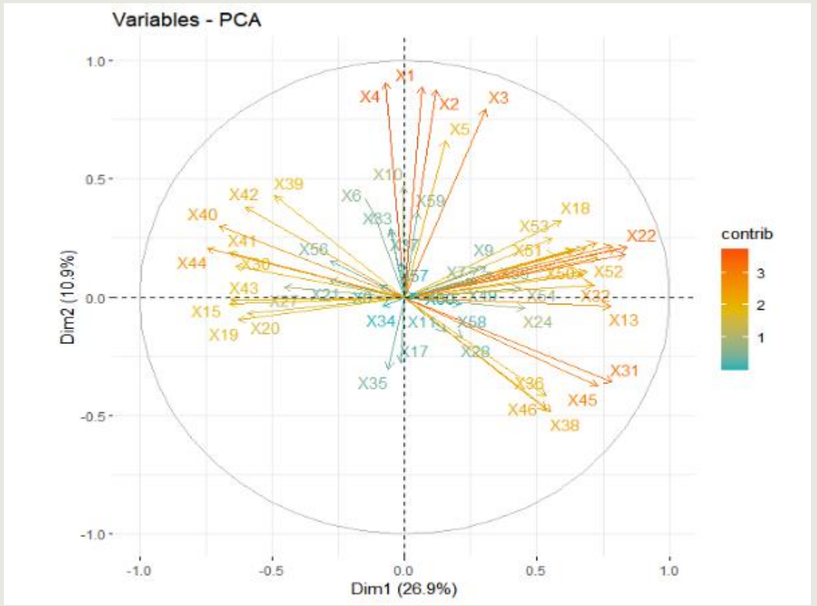
▶ 평가지표: Macro F1-Score



<X11 ~ X16의 산점도, class를 색상으로 구분하여 표시>

Data 탐색 - PCA 이용

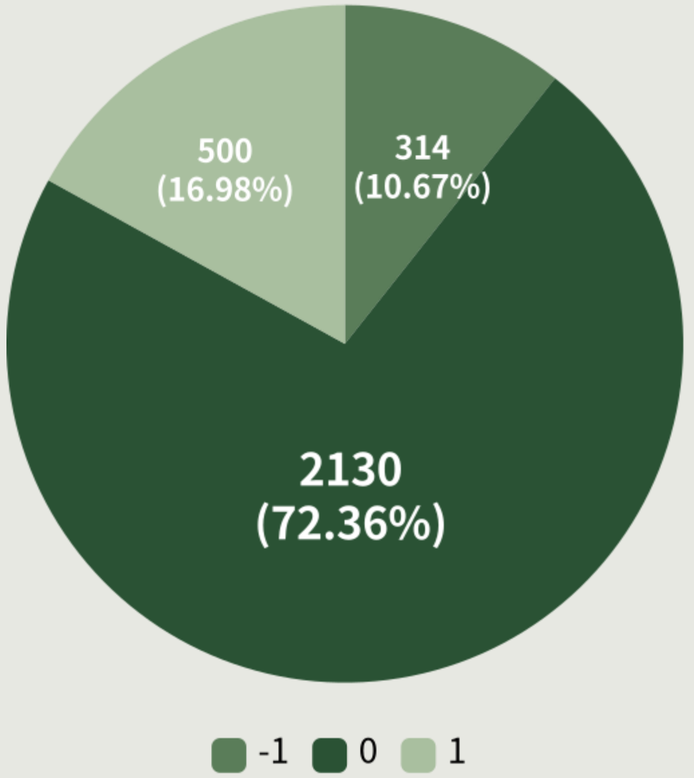
: Data 시각화를 목적으로 PCA를 진행



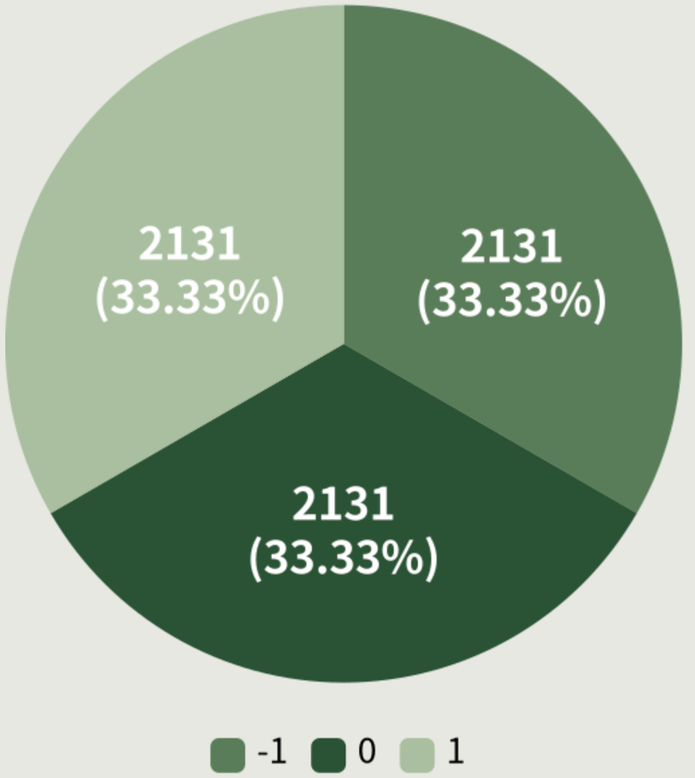
- 첫 번째 주성분: 수익성, 자산비율 등
- 두 번째 주성분: 규모 지표

- 반응변수인 "C"값을 기준으로 grouping
- 수익성, 자산비율이 높을수록 "C"가 1인 경향

Class Imbalance



SMOTE



※ 입력된 자료의 개수와 비율은 전체데이터(trainset.csv) 기준

Data Transformation

01 "Y_b" 변수형을 문자형에서 숫자형으로 변환

- "Y_b" 변수의 변수값은 'D', 'C', 'CC', 'CCC-', 'CCC', 'B-', 'B', 'B+', 'BB-', 'BB', 'BB+', 'BBB-', 'BBB', 'BBB+', 'A-', 'A', 'A+', 'AA-', 'AA', 'AA+', 'AAA' 로 구성 (총 21개의 값 가짐.)
- model에 따라 "Y_b"를 factor형을 처리하는 방식이 다름. 이를 위한 절차의 간편성을 위해 level 순서대로 0~20의 숫자에 대응.

D	C	CC	CCC-	CCC	B-	B	B+	BB-
0	1	2	3	4	5	6	7	8

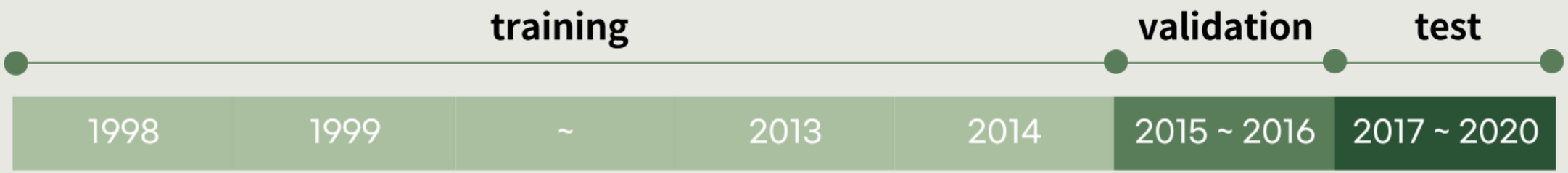
BB	BB+	BBB-	BBB	BBB+	A-	A	A+	AA-
9	10	11	12	13	14	15	16	17

AA	AA+	AAA
18	19	20

02 반응변수 'C'를 순서형으로 변환

- -1, 0, 1로 구성되어 있는 변수인데, 순서가 있는 범주형 변수이므로 순서형으로 변환.

훈련, 검증데이터 분할



목적 : 많은 모델 후보 중 가능성있는 후보를 추려내기 위해 검증데이터 필요

01 시간의 순서에 따라 분할

- 시계열 데이터의 특성 상 훈련데이터보다 미래 시점의 데이터를 검증데이터로 사용하는 것이 적절함.
- 테스트데이터와 시점상 가까울 수록 더 유사한 정보를 담고 있을 것으로 생각
- 최대한 많은 훈련데이터를 확보하기 위해 검증데이터를 400개 정도만 확보 (2년)

02 두 가지 SMOTE 데이터 준비

- 정확한 검증데이터 오류를 측정하기 위해서는 훈련데이터에 검증데이터의 정보가 포함되면 안됨.
- perfect_data.csv : 검증데이터 오류를 측정하기 위한 훈련데이터의 SMOTE 데이터
- final_perfect_data.csv : 최종 모델 훈련을 위한 전체데이터 SMOTE 데이터

설명변수 조합 선택 및 나이브벤치마크

- multinomial logistic regression

✓ 범주가 3개 이상(multi)이며 명목형(nomial)일 때 사용한다.

$$\Pr(Y_i = 1) = \Pr(Y_i = K)e^{\beta_1 \cdot \mathbf{X}_i}$$
$$\Pr(Y_i = 2) = \Pr(Y_i = K)e^{\beta_2 \cdot \mathbf{X}_i}$$
$$\cdots\cdots$$
$$\Pr(Y_i = K-1) = \Pr(Y_i = K)e^{\beta_{K-1} \cdot \mathbf{X}_i}$$

- 설명변수 조합 선택

- 세 가지 설명변수 조합에 다항 로지스틱회귀를 적합하여 macro f1 score 을 계산
- macro f1 score이 가장 높은 설명변수 조합을 선택
- 고려한 설명변수조합

1) X1~X60 - BX1~BX60, Y_b → macro f1 score = 0.4093

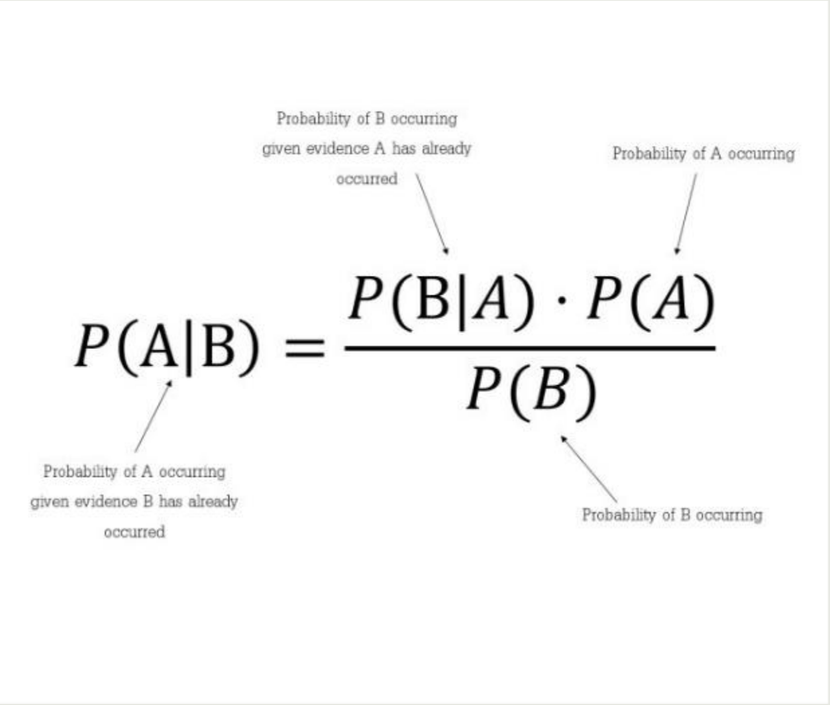
2) X1~X60, Y_b → macro f1 score = 0.5185

3) X1~X60, BX1~BX60, Y_b → **macro f1 score = 0.5469**

➡ 최종 설명변수 조합: 'X1~X60, BX1~BX60, Y_b', 나이브 벤치마크 성능 : 0.5469

Model - Naive Bayes

✓ Naive Bayes는 베이즈 정리를 적용한 분류기



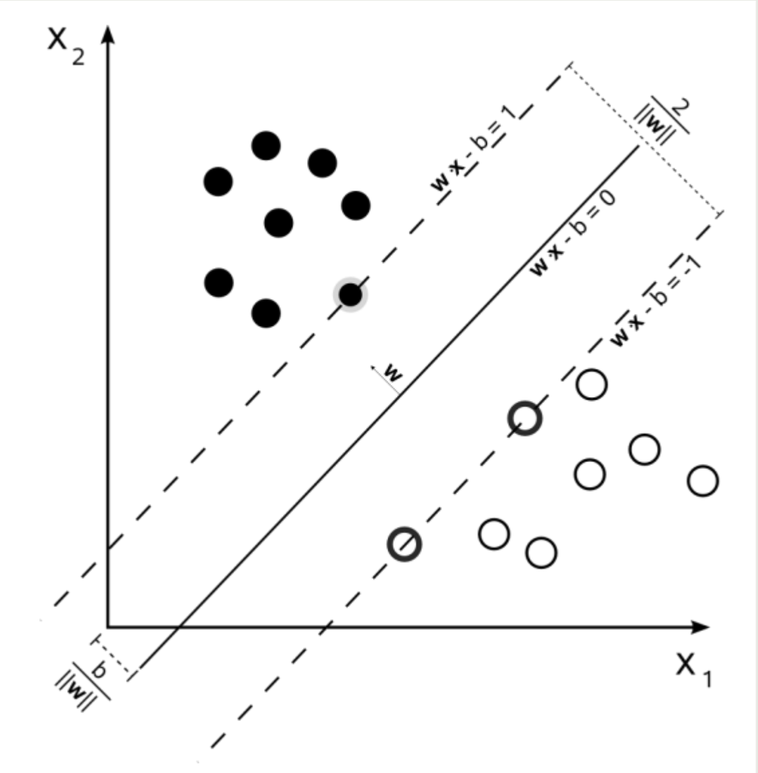
- hyperparameter
  - ✓ **laplace : for laplace smoothing**
    확률이 0일 경우 smoothing 하는 기법
- feature가 확률적으로 '독립'이라는 가정이 전제된다.
  가정에 위반하면 의미있는 결과가 나오지 않을 수 있다.
- feature가 많을 때 **단순화시켜서 쉽고 빠르게 판단**

➡ laplace=0일 때 macro f1 score: "0.4697"

※ macro f1 score는 validation-set 기준

Model - Support Vector Machine

✓ SVM은 분류를 위한 결정경계를 추정하는 모델



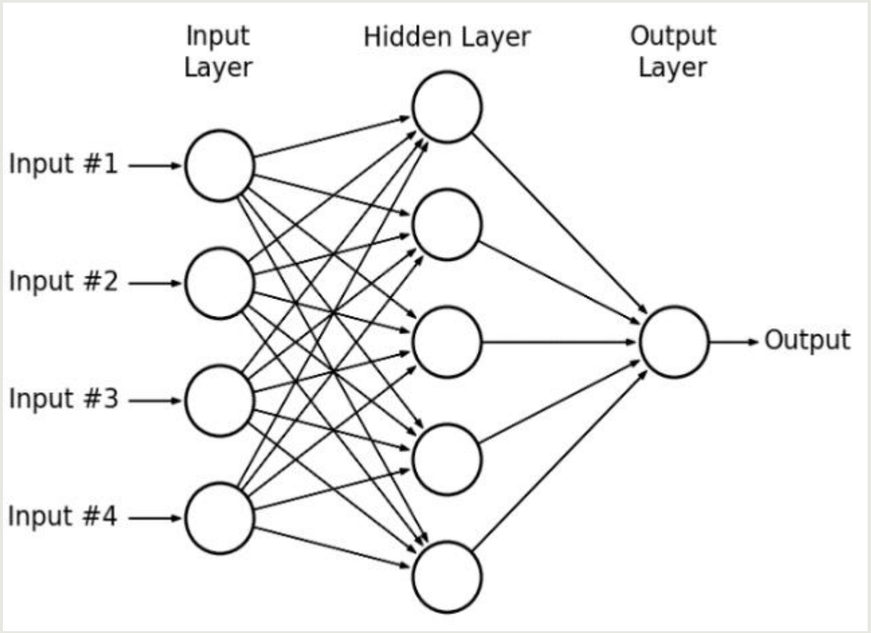
- 분류를 위한 '최적의 결정경계'를 찾는다.
  margine이 최대화되는 경계가 최적의 결정경계다.
- hyperparameter
  - ✓ **cost : 이상치를 얼마나 허용할 것인가** (0~5까지 고려)
    cost 값이 클수록 하드마진(오류허용 안 함)
    cost값이 작을수록 소프트마진(오류허용 함)
  - ✓ **gamma : 결정경계를 얼마나 유연하게 그을 것인가** (0~0.5까지 고려)
    gamma 값이 클수록 구불구불한 경계(오버피팅 가능)
    gamma값이 작을수록 직선에 가까운 경계(언더피팅 가능)
  - ✓ **kernel : 'linear', 'radial', 'sigmoid' 고려**

➡ cost=2, gamma=0.01이며, kernel이 'radial'일 때, macro f1 score: "0.7112"

※ macro f1 score는 validation-set 기준

Model - Deep learning

✓ Deep learning model 중 다층 perceptron을 사용



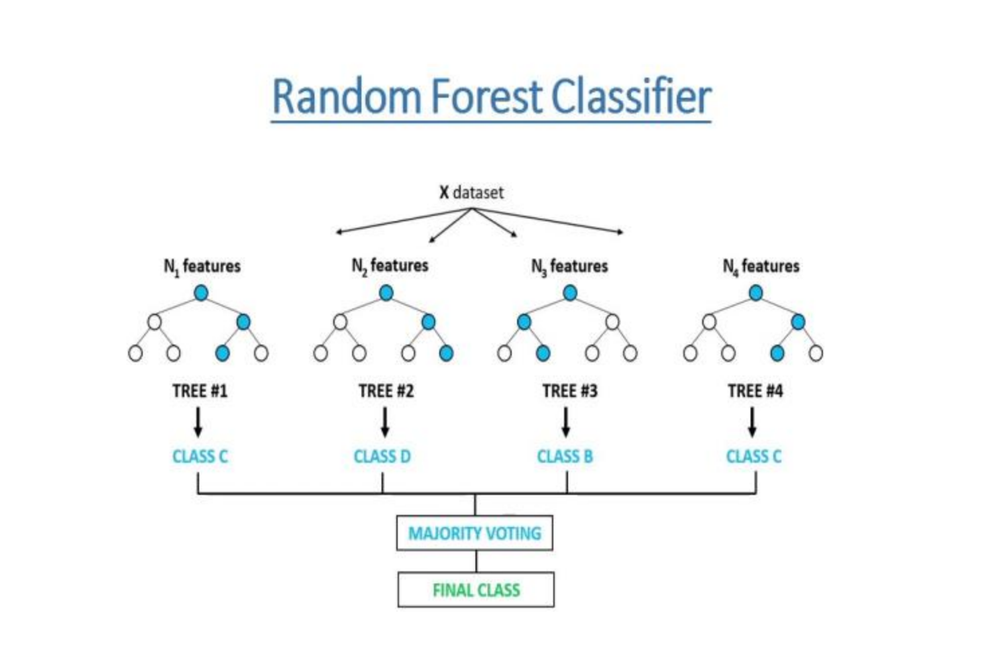
- layer의 개수 & L2 regularization rate값을 다양하게 고려
- 각 layer의 node는 16, 10개
- Output 출력 전, Fully-connected-layer가 아닌 Dropout layer 추가해 overfitting 방지

➡ **Hidden layer가 2개, L2 regularization rate가 0.015 일 때, macro F1-score: "0.6039"**

※ macro f1 score는 validation-set 기준

Model - Random Forest

✓ Tree의 ensemble 기법 중 하나인 Random Forest



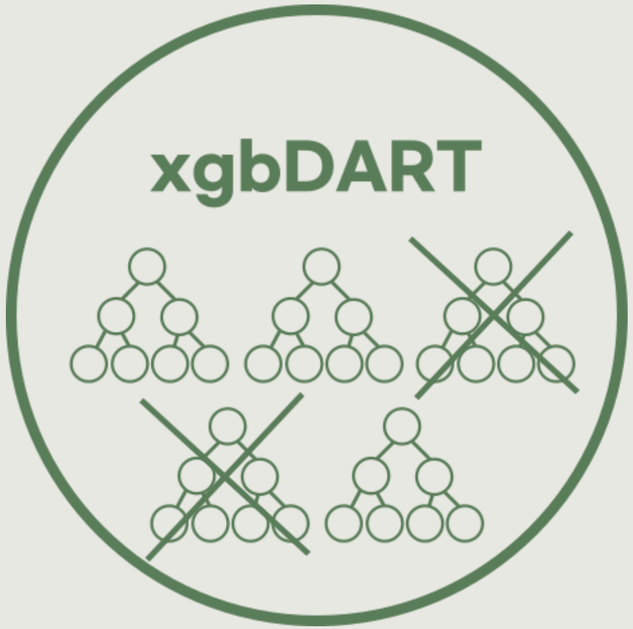
- 최적의 parameter 값 찾기 위해 Grid search 사용
- C=-1, C=1일 때 class_weight 부여
- mtry = sqrt(p) 로 고정

➡ **max_depth : 30, min_samples_leaf : 2, n_estimators : 500 일 때, macro F1-score: "0.5803"**

※ macro f1 score는 validation-set 기준

Xgboost

- Gradient Boosting Model을 병렬처리 및 규제를 가하는 방식으로 속도와 성능면에서 향상시킨 알고리즘.
- model의 type에 따라 기본학습기와 학습 방식이 달라짐. 각 모델 특성에 맞게 하이퍼파라미터 튜닝을 시도.



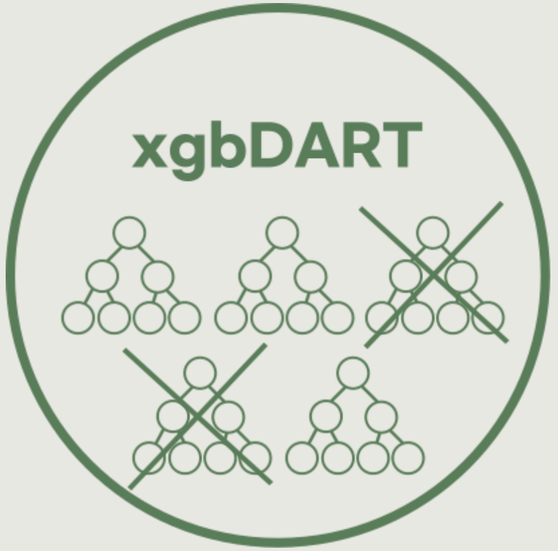
Xgboost - xgbDART

eta = c(0.05, 0.1),
max_depth = c(1,2,3)
nrounds = c(50,100,200)
rate_drop = c(0, 0.25, 0.5)
skip_drop = c(0, 0.25, 0.5)

<1차 튜닝>

subsample = (0.5, 0.7)
gamma = c(0.1, 0.2, 0.3, 0.4)
colsample_bytree = (0.5, 0.7)
rate_drop = c(0, 0.25)
skip_drop = c(0, 0.25, 0.5)

<2차 튜닝>



특징: 기본학습기는 Tree이고 최종 모델에 각 트리를 사용할지 말지를 고려하는 모델

고려사항

- 중요 하이퍼파라미터를 튜닝한 뒤, 2차적으로 세부 하이퍼파라미터를 조정

✓ 세밀한 학습을 위해 낮은 학습률,
✓ 높은 편향을 가진 기본학습기를 위한 낮은 depth,
✓ 적절한 트리의 개수를 찾기 위한 폭 넓은 nrounds,
✓ 과적합 방지를 위한 sample parameter를 고려

validation macro f-score

0.5428

Xgboost - xgbTree

```
eta = c(0.1,0.05)
max_depth = c(2,3)
subsample = c(0.7,0.5,0.3)
nrounds = c(100,200,300,400,500)
gamma = c(0.1,0.2,0.3,0.4)
colsample_bytree = c(0.7,0.5,0.3)
```

특징: 기본학습기는 Tree로 기본학습기를 훈련시킬때 규제를 가하는 모델

고려사항

- ✓ 세밀한 학습을 위해 낮은 학습률,
- ✓ 높은 편향을 가진 기본학습기를 위한 낮은 depth,
- ✓ 적절한 트리의 개수를 찾기 위한 폭 넓은 nrounds,
- ✓ 과적합 방지를 위한 sample parameter를 고려
- ✓ 적절한 규제 정도를 찾기 위한 gamma

validation macro f-score

0.5242

Xgboost - xgbLinear

```
eta = c(0.05, 0.1),
nrounds = c(10,20,30)
lambda = c(0, 0.1,0.3)
alpha = c(0,0.1,0.3)
```

특징: 기본학습기는 규제가 더해진 logistic linear regression model

고려사항

- ✓ 변수간의 교호작용을 고려하기 위해 교호작용항을 추가한 모델(formula : C ~ .^2)과 추가하지 않은 모델(formula : C ~ .)을 비교
- ✓ 세밀한 학습을 위해 낮은 학습률,
- ✓ 적절한 규제 정도를 찾기 위한 lambda, alpha

교호작용항 O
validation macro f-score

0.5106

교호작용항 X
validation macro f-score

0.5129

최종 model 후보선정

- validation set macro f1 score이 높은 4가지 모델을 최종모델후보로 선정

Deep learning
0.6039

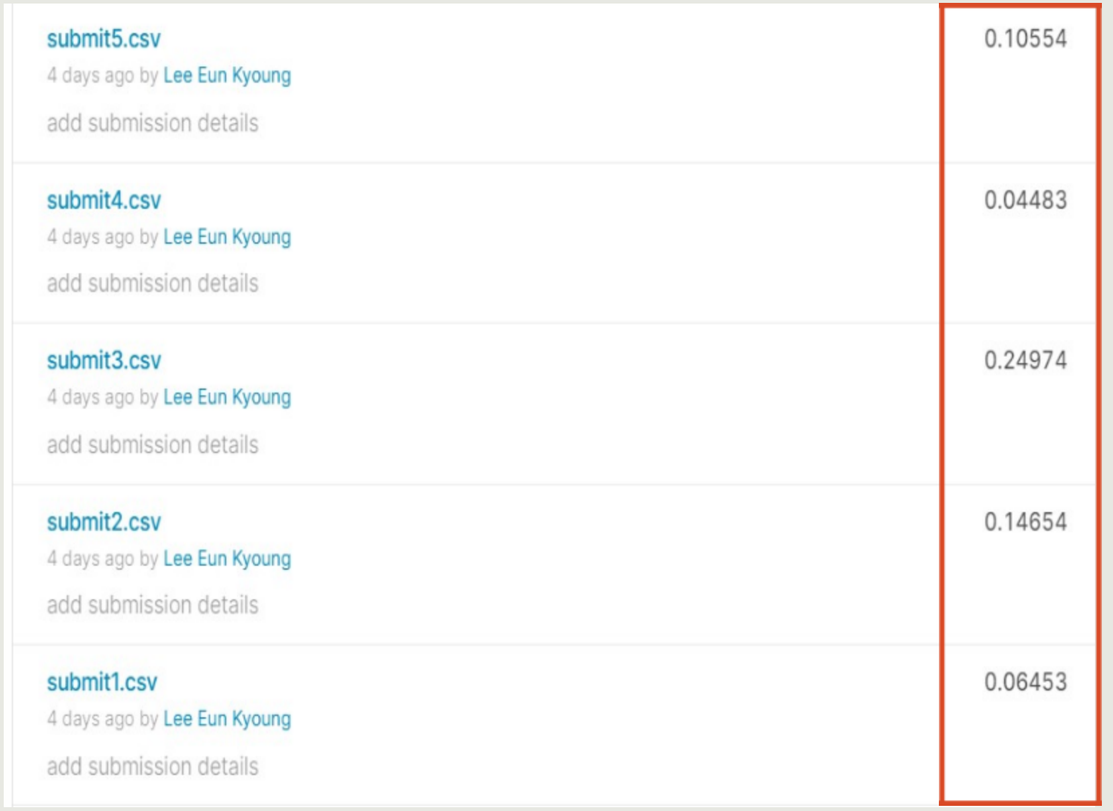
SVM
0.7112

Random Forest
0.5803

Xgboost
0.5428

※ macro f1 score는 validation-set 기준

최종 model 후보선정 - Deep learning 제외



submit5.csv 4 days ago by Lee Eun Kyoung add submission details	0.10554
submit4.csv 4 days ago by Lee Eun Kyoung add submission details	0.04483
submit3.csv 4 days ago by Lee Eun Kyoung add submission details	0.24974
submit2.csv 4 days ago by Lee Eun Kyoung add submission details	0.14654
submit1.csv 4 days ago by Lee Eun Kyoung add submission details	0.06453

- 검증데이터에서 좋은 성능을 가진 하이퍼파라미터 조합으로 전체데이터를 훈련 후 test data예측.
- 최고 macro f1-score : 0.24974
- test-set에서는 딥러닝이 좋은 성능을 내지 못함.
- Dataset의 개수가 적어 feature들을 잘 학습하지 못한 것으로 예상됨.

최종 model 선정

- 최종 모델 후보에서 Deep Learning model를 제외한 3개의 모델을 최종 모델 후보로 선정

SVM

Test macro f1-score:
0.51108

Random Forest

Test macro f1-score:
0.51774

Xgboost

Test macro f1-score:
0.49728

※ macro f1 score는 test-set 기준

최종 model

SVM

Test macro f1-score:
0.51108

-1	0	1
26	709	43

+

Random Forest

Test macro f1-score:
0.51774

-1	0	1
92	663	23

+

Xgboost

Test macro f1-score:
0.49728

-1	0	1
65	694	19

```
mix_pred = svm.pred + rf.pred + xgb.pred
mix_pred = ifelse(mix_pred <= -0.5, -1, ifelse(mix_pred >=0.5, 1, 0))
```

➡ 최종 test data macro f1-score : 0.53480

프로젝트 보완점

01 SMOTE 외의 클래스 불균형 처리 방법 적용

- 본 프로젝트에서는 클래스 불균형 처리 방법을 SMOTE 방법만 고려함
- 다른 클래스 불균형 처리 방법을 사용한 데이터와 결과를 비교해봐도 좋을 듯 함

02 각 모델별로 잘못 예측된 결과를 분석 후 개선

- 분류의 결과에서 오분류된 데이터를 살펴보고 어떤 데이터가 잘 분류되지 않았는지 파악