2024.02.03

14기 신입교육세션

5 주 차 - 모 델 링 2

B.a.f

CONTENTS

1. 모델링 과제 피드백

[1] 랜덤포레스트

[2] GBM

[3] 선형회귀

[4] 주요 피드백 내용

B.a.f

랜덤포레스트 (Random Forest)

: 앙상블 기법 중 '배깅'에 해당하며, 여러 개의 Decision Tree를 만들어 학습하고 각 모델의 평균값으로 결과를 산출하는 모델

< 장 점 >

- 단일 Decision Tree 모델에 비해 성능이 뛰어남
- 매개변수(parameter)를 많이 튜닝하지 않아도 잘 작동함
- 데이터 스케일링 필요 X

< 단 점 >

- Decision Tree에 비해 모델이 복잡해짐
- Text 데이터와 같이 고차원이고 희소한 데이터에서는 잘 작동하지 않음

  => 선형모델에 비해 더 많은 메모리와 훈련시간이 필요함

GBM (Gradient Boosting)

: 앙상블 기법 중 '부스팅'에 해당하며, 이전 예측기가 만든 오차를 보완하도록 예측기를 순차적으로 추가하는 모델

!! 가중치 업데이트로 '경사하강법(Gradient Descent)'을 이용

< 장 점 >

- 비교적 높은 정확도를 가짐
- 데이터 스케일링 필요 X

< 단 점 >

- 매개변수(parameter) 조정을 잘 해줘야함
- 결과를 이해하고 해석하기 어려움
- 계산량이 많아서 훈련시간이 다소 김
- Text 데이터와 같이 고차원이고 희소한 데이터에는 잘 작동하지 않음

선형 회귀 (Linear Regression)

: 종속변수 y와 1개 이상의 독립변수 X의 선형상관관계를 모델링하는 회귀분석 기법

!! 주로 최소제곱법(Ordinary Least Square)을 통해 오차를 최소화하는 회귀 모델을 추정

< 장 점 >

- 간단하고 빠름
- 이해하기 쉬움
- 조정해야할 매개변수(parameter) 수가 적음

< 단 점 >

- 이상치에 민감
- 데이터 스케일링의 영향을 받음
- 데이터 전처리 과정이 많이 요구됨

주요 피드백 내용

(1) train셋과 동일하게 test셋도 전처리해주기

(2) 원핫인코딩 시, pd.get_dummies() 보다 sklearn의 OneHotEncoder() 사용 권장

!! pd.get_dummies()는 train셋에는 있지만 test셋에는 없는 범주를 더미변수로 나타낼 수 없음

```
train
```

	num1	num2	cat1
0	1	10	a
1	2	20	a
2	3	30	b
3	4	40	c
4	5	50	c

```
pd.get_dummies(train)
```

	num1	num2	cat1_a	cat1_b	cat1_c
0	1	10	1	0	0
1	2	20	1	0	0
2	3	30	0	1	0
3	4	40	0	0	1
4	5	50	0	0	1

```
test
```

	num1	num2	cat1
0	1	10	a
1	2	20	a
2	3	30	b
3	4	40	b
4	5	50	a

```
pd.get_dummies(test)
```

	num1	num2	cat1_a	cat1_b
0	1	10	1	0
1	2	20	1	0
2	3	30	0	1
3	4	40	0	1
4	5	50	1	0

2. 분류 & 회귀

[1] 지도 학습

[2] 분류 문제

[3] 회귀 문제

B.a.f

지도 학습

지도 학습이란 ?

종속 변수 y가 데이터에 있는 경우, y를 예측하기 위한 학습 방법

좋은 모델이란 ?

test data

train data로 학습한 모델이, 새로운 데이터가 주어져도 정확히 예측하는 것

- 일반화
- 과대적합 (overfitting) & 과소적합 (underfitting)

분류 문제

binary

- 이메일이 피싱 메일은 아닐까 ?
- 고객이 제품을 계속 사용할까 ?
- 사용자가 광고를 클릭할까 ?

☑ 타이타닉 데이터는 이진 분류 문제

categorical

- 고객의 대출 등급은 무엇일까 ?
- 사용자가 제일 좋아하는 음악 장르는 무엇일까 ?

모델 종류

- 로지스틱 회귀, SVM, 랜덤포레스트, XGBoost 등

분류 문제

혼동행렬

Predicted

Actual		Positive	Negative
	Positive	TP (True Positive)	FN (False Negative)
	Negative	FP (False Positive)	TN (True Negative)

- TP : 모델이 Positive라고 예측한 것이 정답인 샘플
- FP : 모델이 Positive라고 예측한 것이 오답인 샘플 (1종 오류)
- FN : 모델이 Negative라고 예측한 것이 오답인 샘플 (2종 오류)
- TN : 모델이 Negative라고 예측한 것이 정답인 샘플

분류 문제

Accuracy

Predicted

Actual		Positive	Negative
	Positive	TP (True Positive)	FN (False Negative)
	Negative	FP (False Positive)	TN (True Negative)

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- 전체 샘플 중 정답을 맞춘 비율
- Accuracy만 가지고 성능을 판단해서는 안됨
- 불균형 데이터에서는 Accuracy로 성능 판단 X

  ex) 100명 중 1명이 암환자인 데이터
  샘플 모두 음성이라 예측해도 accuracy는 99%

분류 문제

Precision (정밀도)

Predicted

Actual

	Positive	Negative
Positive	TP (True Positive)	FN (False Negative)
Negative	FP (False Positive)	TN (True Negative)

$$\text{Precision} = \frac{TP}{TP + FP}$$

- True라 예측한 것중 진짜 True인 비율
- Precision이 높다 : 정말 확실한 경우에만 참이라 예측
- Precision이 낮다 : 참이 아닌데 참이라 예측한 샘플 수가 많다

  ex) 스팸메일이 아닌데 스팸메일이라 판단해 차단함

분류 문제

Recall (재현율)

Predicted

Actual		Positive	Negative
	Positive	TP (True Positive)	FN (False Negative)
	Negative	FP (False Positive)	TN (True Negative)

$$\text{Recall} = \frac{TP}{TP + FN}$$

- 실제 True 샘플 중 True라 예측한 비율
- Recall이 높다 : True라 예측한 샘플이 많다
- Recall이 낮다 : True인데 못찾은 샘플이 많다

분류 문제

f1-score

Predicted

Actual		Positive	Negative
	Positive	TP (True Positive)	FN (False Negative)
	Negative	FP (False Positive)	TN (True Negative)

Precision

Recall

Trade off ⟫

$$F1\ Score = 2 \times \frac{recall \times precision}{recall + precision}$$

회귀 문제

regression

- 다음 달 전력사용량은 얼마일까?
- 설날 선물 수요량은 얼마일까 ?
- 영화를 보러 몇 명이 올까 ?

☑ 과제로 나가는 따릉이는 회귀 문제

모델 종류

- Linear Regression, SVR, 랜덤포레스트, XGBoost 등

회귀 문제

R^2 : 결정계수

- 실제 관측값의 분산대비 예측값의 분산을 계산
- 0~1까지 나타낼 수 있고, 1에 가까울수록 설명력을 높게 가지는 모델

MSE (Mean Squared Error)

- 종속 변수와 단위가 다름
- 에러를 제곱하기 때문에 이상치에 민감

MAE (Mean Absolute Error)

- Error에 절대값을 취해 Error의 크기를 그대로 반영
- 예측변수와 단위가 같고 직관적임
- MSE보다 이상치에 robust함

RMSE

- MSE에 루트를 취한 값
- 종속 변수와 단위가 같음

수식

$$R^2 = \frac{\sum_{i=0}^{n}(Y_i - \hat{Y}_i)^2}{\sum_{i=0}^{n}(Y_i - \bar{Y}_i)^2} = 1 - \frac{SSE}{SST} = \frac{SSR}{SST}$$

$$MSE = \frac{1}{n}\sum_{i=0}^{n}(Y_i - \hat{Y}_i)^2$$

$$MAE = \frac{1}{n}\sum_{i=0}^{n}|Y_i - \hat{Y}_i|$$

3. 모델 학습

[1] 모델 선택

[2] 하이퍼파라미터 튜닝

[3] 차원 축소

B.a.f

모델 선택

1. 모델의 목표에 맞춰 모델 후보를 설정

- 성능 개선
- 학습 속도

(예시)

Decision Tree

- 구조가 단순하여 해석이 쉬움
- 분류,회귀 모두 사용 가능

Random Forest

- 대용량 데이터에 효과적
- 과적합 문제도 최소화하여 모델 정확도 향상

XGBoost

- 병렬처리로 학습, 분류 속도가 빠름
- 과적합 방지 가능

2. 모델 최적화 후 모델 목표에 제일 맞는 모델 선정

하이퍼파라미터 튜닝

하이퍼파라미터

모델의 동작 및 학습 과정을 제어하는 매개변수

》 **쉽게 말해 우리가 직접 조정할 수 있는 값들**

왜 해야할까?

같은 모델을 사용해도 하이퍼파라미터 값들에 따라 모델의 성능이 달라짐 !
따라서 모델을 최적화시키기 위해 튜닝은 필수적

튜닝 방법

- Grid Search
- Manual Search

하이퍼파라미터 튜닝

Manual Search

사용자의 직관이나 경험으로 하이퍼파라미터를 조정하여 사용

1. 먼저 **임의의 값**을 대입해 결과를 살핀다.
2. 그 결과에 따라 **값을 조정**해가며 변화를 관찰한다.
3. 값을 하나씩 대입해보고 조정하는 과정을 반복하면서 **최적의 값**을 찾는다.

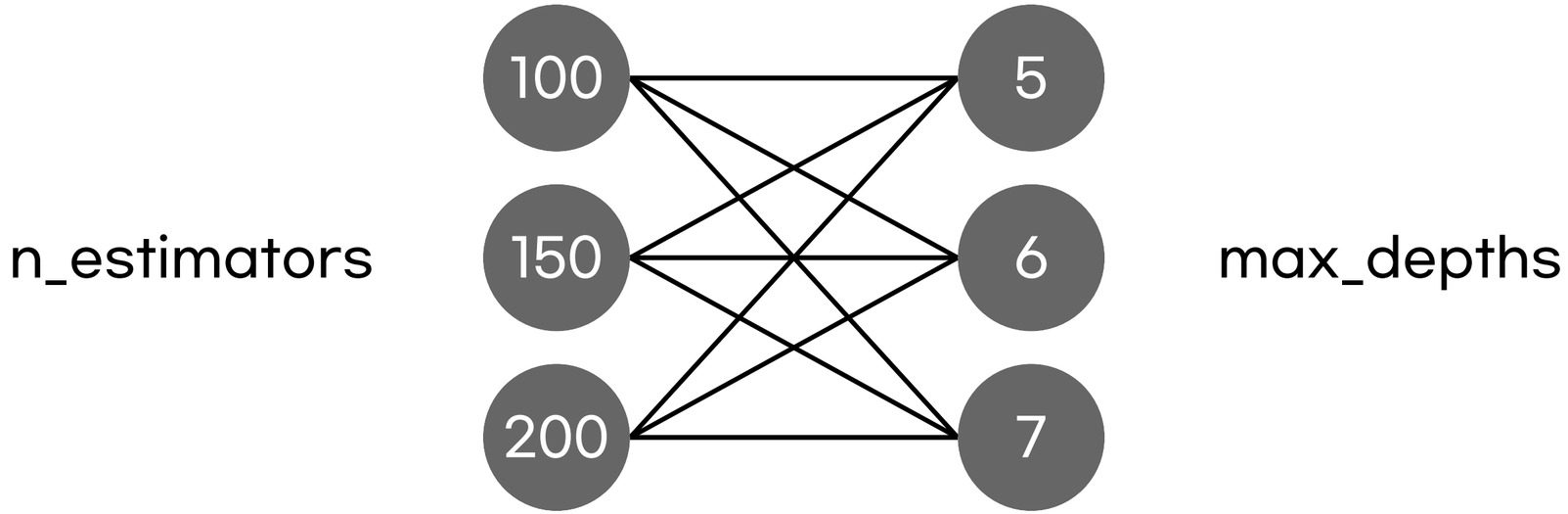
매우 단순하고 쉬운 방법이지만
그만큼 최적의 파라미터 값들과 조합을 찾는 것이 힘들다

하이퍼파라미터 튜닝

Grid Search

우리가 지정한 하이퍼파라미터들의 후보군들의 조합 중 Best 조합을 선별

(예시)



라이브러리가 존재하여 사용이 간편하지만

조합이 늘어갈 때마다 시간 소요가 크다는 단점

최종 변수 선택

변수를 선택해야하는 이유 ?

1. 종속 변수 예측에 영향을 주지 않는 경우
2. 독립변수들끼리 다중공선성이 발생한 경우

》 정확한 예측을 위해 적절한 변수 선택 또는 PCA 같은 차원 축소가 필요

(1) 변수 선택 방법

- 전진선택법 (forward selection)
- 후진선택법 (backward selection)
- 단계선택법 (stepwise method)
- 변수중요도를 보고 판단
- 변수의 정의를 보고 판단

(2) 차원축소

- PCA
- FA
- MDS

4. 모델 평가

[1] 평가지표

[2] 변수 중요도

[2] 과적합 & 과소적합

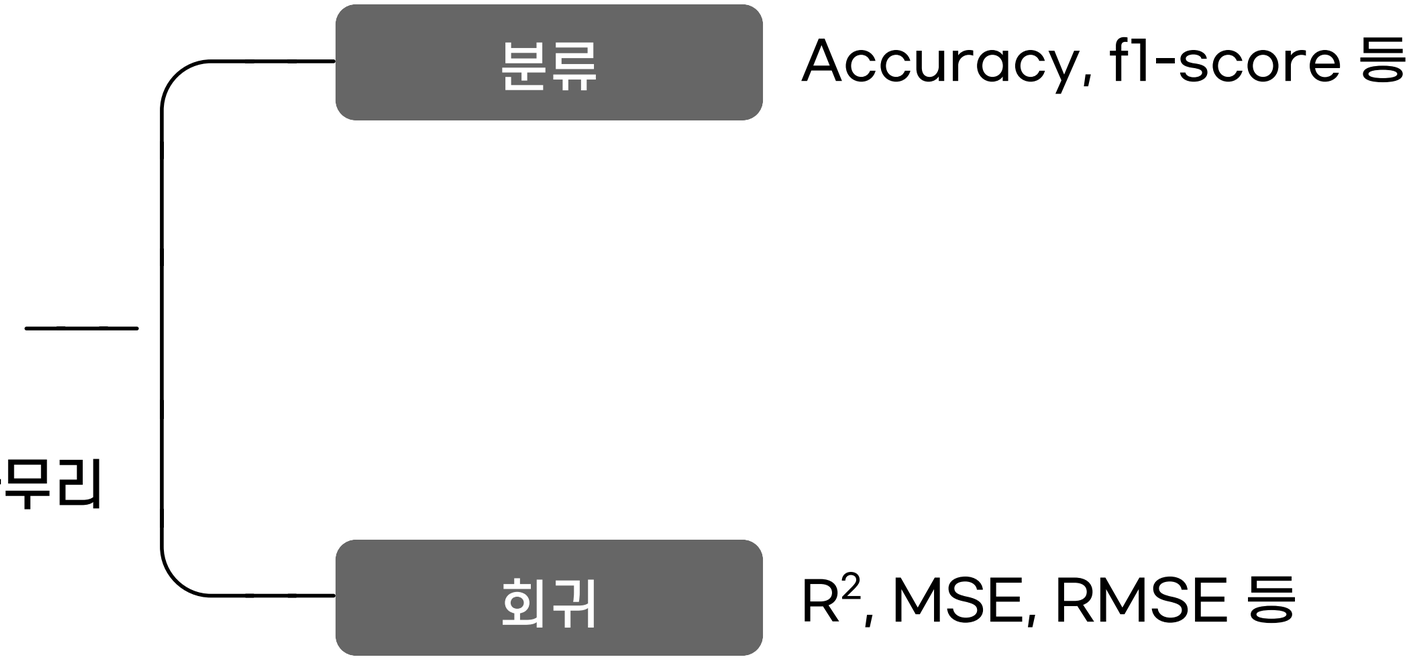
B.a.f

평가지표

우리는 Test셋의 종속 변수 값을 알 수 없는 경우가 대부분
따라서 valid셋을 활용하여 성능 개선

(모델링 순서)

1. 모델 선언
2. 모델 학습
3. valid 셋을 활용하여 성능 확인
4. 3번 과정을 반복하여 모델 최적화
5. 최종 모델로 test셋을 예측하며 마무리

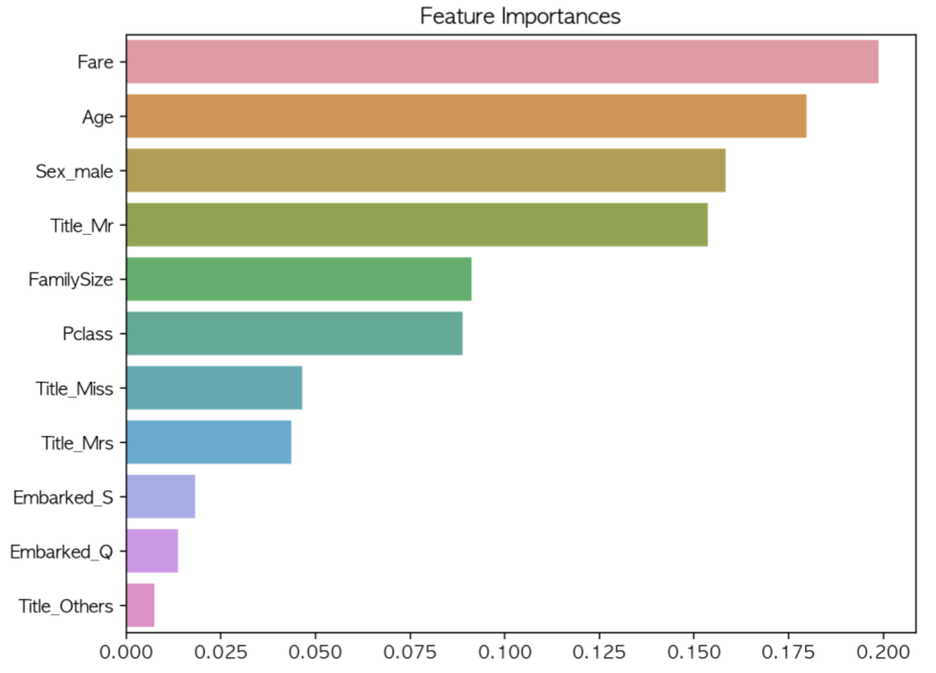


**종속변수에 스케일링 시 값 복원 후 평가지표 확인

변수 중요도

머신러닝 모델

: 머신러닝 모델들은 변수 중요도를 알 수 있음 ** Linear Regression - 회귀 계수와 변수가 유의한지 확인
ex) RandomForest, XGBoost, Decision Tree, Lgbm 등

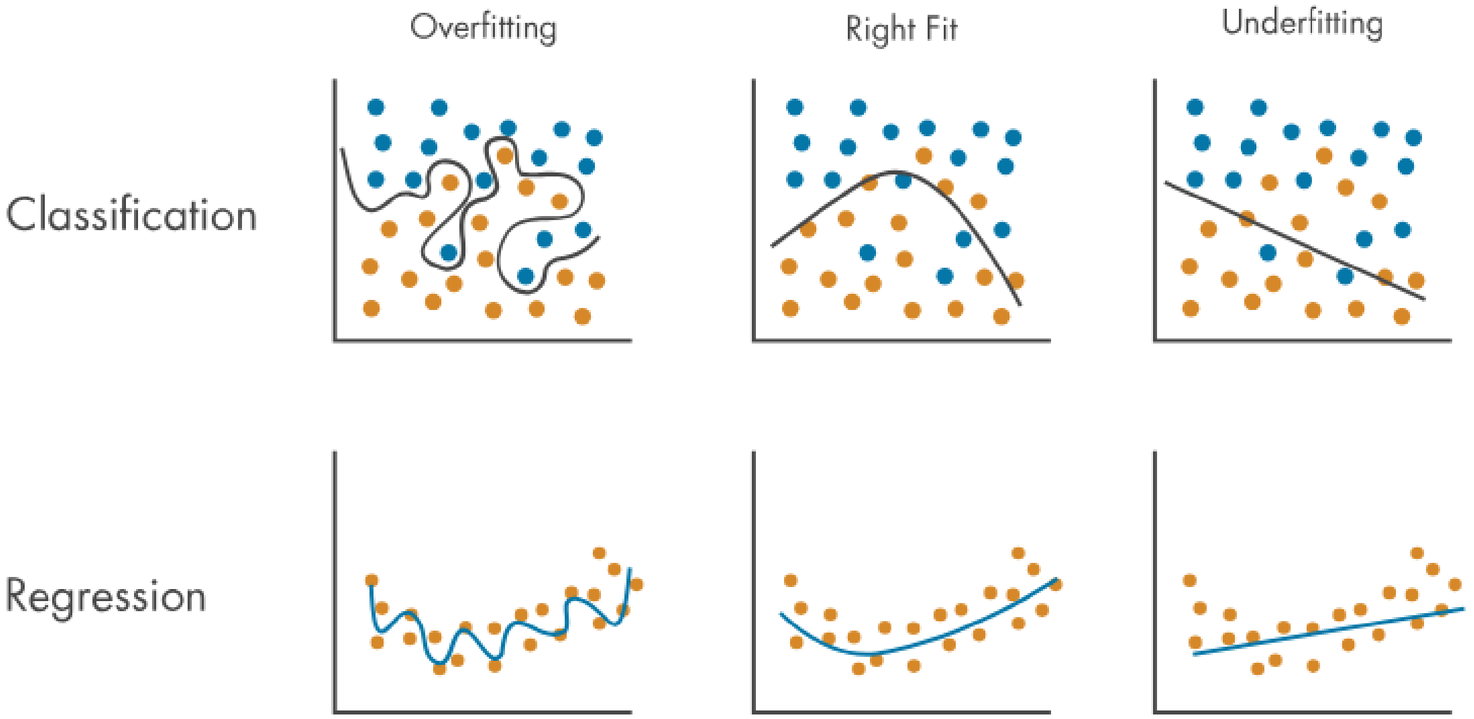


변수중요도를 보고 해석 가능
분석 목표에 맞는 기대효과와 해결방안 생각 가능
데이터 분석가에게 중요한 역할

과적합 & 과소적합

분류문제

회귀문제



모델의 학습이 지나치게 train data에 맞추어져 일반화 성능이 떨어짐

train data에 대해 제대로 학습되지 않음

과적합 & 과소적합

과적합 방지 방법

(1) 결과에 영향을 덜 주는 변수를 드랍하며 모델을 더 간단하게 만들어주기
(2) 변수 정규화
(3) valid dataset으로 일반화를 위해 시도 ex) K-fold

5. 실습 & 과제 안내

[1] 실습 진행

[2] 과제 안내

B.a.f

실습 진행

1. 모델 선정

2. 모델 학습

3. 하이퍼파라미터 튜닝

4. 최종모델 선정

5. valid셋으로 성능 확인 + 변수 중요도 확인

6. test셋에 대한 예측 후 csv로 저장

과제 안내

1. 모델 선정
따릉이 데이터셋에 맞는 모델 후보군 하나 선정해보기 - 근거 생각

2. 모델 학습
해당 모델의 기본 모델로 학습 후 성능 확인

3. 하이퍼파라미터 튜닝
해당 모델의 하이퍼파라미터 튜닝해보기

4. 최종모델 선정
모델과 하이퍼파라미터, 변수 선택까지 진행

5. 최종 모델에 대한 성능 + 변수 중요도 확인 후 해석까지

6. test셋에 대한 예측 후 csv로 저장

감사합니다

B.a.f