
Kaggle 주택 가격 예측

김지선

Contents

Part1 - 이론

- 회귀 분석
- 데이터 전처리
- 성능평가

Part2 - 실습

- 캐글 흐름
- 주택가격 실습 (Kaggle)
- 데이터 분석
- 알고리즘의 종류 (사용 목적과 용도)

Part 1 - 이론

1. 회귀의 분류

- 규제가 없는 회귀
 - 단순 선형 회귀
 - 다항 회귀
 - 다중 회귀
- 규제가 있는 회귀
 - 릿지 회귀
 - 라쏘 회귀
 - 엘라스틱넷

회귀 수식에서 용어 정리

- 회귀 수식에서 나오는 X, Y 는 아래와 같은 의미를 가진다.

변수	뜻
X	독립 변수(설명 변수), 입력 값
Y	종속 변수(목적 변수), 예측하려는 정답

회귀의 종류

- 단순 선형 회귀

- 직선으로 데이터를 표현

$$y = wx + b$$

* w(weight): 가중치(회귀 계수)

b(bias): 편향

- 다중 회귀

- 독립 변수 x(2개 이상)를 고려하여 정답 y를 예측

$$y = w_1x_1 + w_2x_2 + \dots + w_nx_n + b$$

- 다항 회귀

- 곡선으로 x(1개에 대한 차수의 증가)와 y간의 관계를 표현

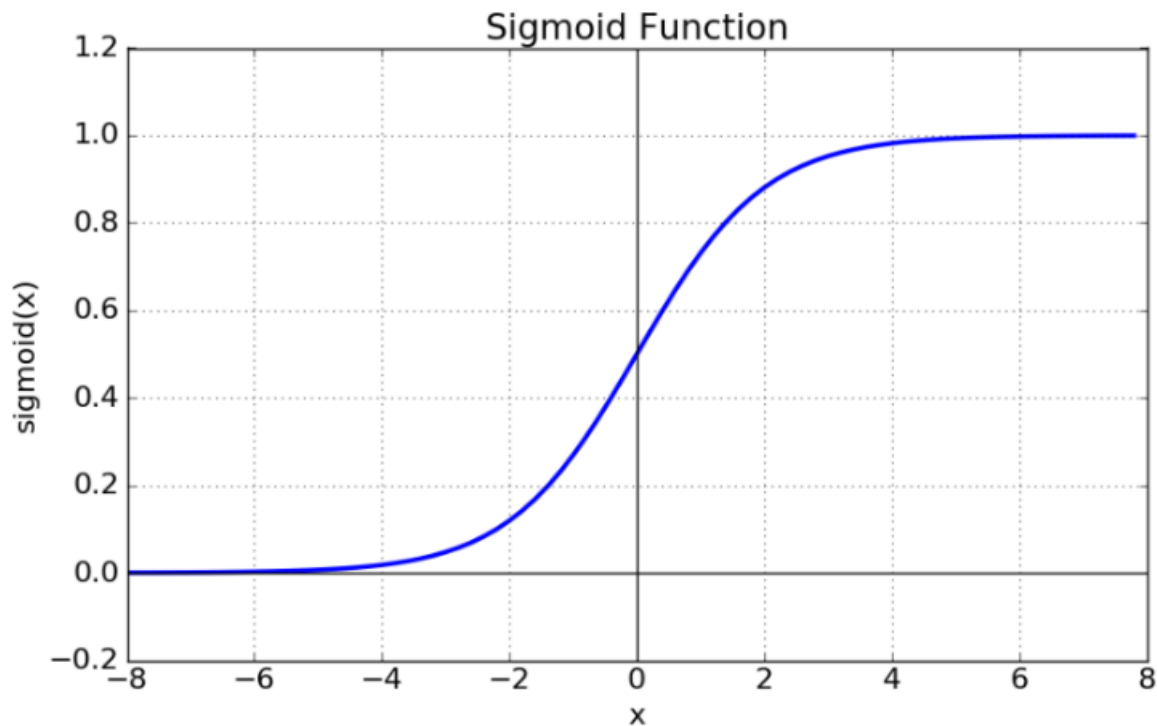
$$f(x) = w_nx^n + w_{n-1}x^{n-1} + \dots + w_2x^2 + w_1x + b$$

1.3 회귀의 분류

시그모이드 함수(로지스틱 함수)

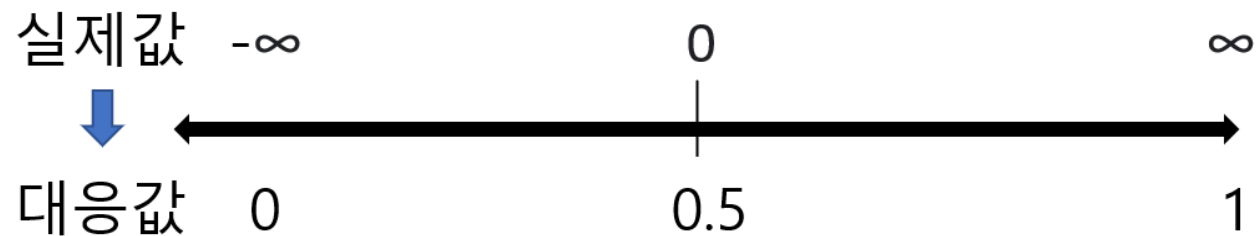
- 시그모이드 함수의 대표적인 함수가 로지스틱 함수여서 혼용되어 사용됨

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



Sigmoid 함수의 역할

- 입력 값을 0에서 1사이의 값으로 정규화해줌
- 예시
 - 사이킷런 `LogisticRegression()` : 이름은 회귀이지만 분류 알고리즘
 - 다중 회귀 수식을 사용하여 y 값을 시그모이드함수에 입력으로 넣어 정규화된 값(확률)으로 출력




Sigmoid 활용 예시 - LogisticRegression

- LogisticRegression 알고리즘
 - 클래스에 대한 확률 값을 기반으로 예측함

- Step1
 - Test feature가 입력으로 들어감

$$y = w_1x_1 + w_2x_2 + w_3x_3 + b$$



꽃잎의 색상	꽃잎의 너비	꽃잎의 길이	꽃의 종류
3.4	3.2	2.3	?
1.5	4.3	2.2	?

- Step2
 - Y 값을 시그모이드에 입력해 값을 정규화하고 0.5를 기준으로 분류

2. 규제가 있는 회귀 - L1/L2 규제

- 예측 데이터의 오류 최소화 + 회귀 계수 크기 제어
 - ⇒ 즉 회귀 계수가 **너무 커지지 않도록 규제**하는 것
- 규제 – 과적합을 방지하기 위해 사용
 - L1 규제 – 규제가 커질수록 가중치 값이 0에 가까워짐
 - L2 규제 – 규제가 강해져도 과소적합이 심해지진 않음
 - 과소적합 – 한 값에 회귀계수가 너무 작아지는 것
 - 과대적합 – 한 값에 회귀계수가 너무 커지는 것

릿지(Ridge) 회귀

- L2 규제로 회귀 계수가 큰 값의 영향 감소하기 위해
그 큰 값을 줄여줌
- 객체=Ridge(alpha=k)
 - 하이퍼 파라미터 k로 규제의 정도를 조정
 - K와 규제는 반비례
- 객체.fit(train_feature,train_target)

라쏘 회귀

- L1 규제를 선형 회귀에 적용
- Feature selection
- 회귀 계수를 없애는 것이 아닌 영향력 약한 회귀 계수를 제거
- 객체=Lasso(alpha=k)
 - 하이퍼 파라미터 k로 규제의 정도를 조정
 - K와 규제는 반비례
- 객체.fit(train_feature,train_target)

엘라스틱 넷 (Elastic Net)

- L1 규제와 L2 규제를 결합한 회귀 모델
- 객체=ElasticNet(alpha, l1_ratio)
 - Alpha= L1 alpha+ L2 alpha
 - l1_ratio=a / (a+b)

정리

- 단항 회귀
- 다항 회귀
- 다중 회귀
- 로지스틱
- 릿지
- 라쏘

데이터 전처리

- 수치형(Numerical Data) 데이터(int, float) -> 전처리 필요 X
- 범주형 데이터(Categorical Data) -> 전처리 필요 O
 - 원 핫 인코딩
 - 라벨 인코딩
 - Feature Vectorization (Bag of word~)
 - 임베딩 (문자를 단어로 표현, Word2Vec)
- sklearn의 **ML 알고리즘은 문자열 데이터를 입력 값으로 받지 못합니다**

3. 범주형 데이터 전처리

- 각 피처가 가지는 값들의 숫자 범위(Scale)가 다를 경우 이 값의 범위를 일정한 범위로 맞추는 작업
- 트리계열을 제외한 대부분의 머신러닝 알고리즘들이 피처의 스케일에 영향을 받는다.
 - 선형모델, SVM 모델, 신경망 모델 등
-

4. 수치형 데이터 전처리 (Overview)

- 스케일링
 - 정규화
 - 표준화
- 결측치 처리

라벨 인코딩

- 문자별로 들어간 코드 값을 숫자형으로 매핑하는 것
- 같은 공간에 더 많은 정보 표현 가능
- [TV, 냉장고, 전자레인지] -> [0,1,2]

TV	냉장고	전자레인지
----	-----	-------



0	1	2
---	---	---

라벨 인코딩

- 사용 방법

```
From sklearn.preprocessing import LabelEncoder
```

```
객체=LabelEncoder( )
```

```
객체.fit(카테고리 리스트)
```

```
Labels=encoder.transform(카테고리 리스트)
```

원 핫 인코딩

- 한 칼럼에 1/0으로 값을 나타내는 것
- 희소 행렬로 공간을 많이 차지하지만 관계성 방지 가능

color	red
color	green
color	blue
color	red

one-hot
encoding →

color_red	color_green	color_blue
1	0	0
0	1	0
0	0	1
1	0	0

원 핫 인코딩

- From sklearn.preprocessing import OneHotEncoder
- Import numpy as np
- 객체=OneHotEncoder()
- 객체.fit(카테고리 리스트)
- 객체.transform(카테고리 리스트)

결측치 처리

- Feature값 Null값의 비율이 **매우 작을** 경우
 - ⇒ 단순히 평균값, 최빈값 등으로 대체하는 방법
- 중요한 feature의 Null값 분포가 **일정 수준 이상**일 경우
 - ⇒ 위 경우가 Null값 처리의 핵심
 - ⇒ 중요한 피처인 경우 위 방식으로 처리시 예측 왜곡이 발생할 수 있음
 - ⇒ 데이터를 상세히 검토하여 더 정밀한 값으로 대체해야함
- Feature값 **대부분이 Null값인** 경우
 - ⇒ 해당 피처를 드롭하는 것이 ML 알고리즘 성능개선에 효과적

스케일링의 필요성

- 서로 다른 변수의 값 범위를 일정한 수준으로 맞추는 작업
- 사용 이유
 - 변수의 크기가 모델학습에 영향을 끼치는 경우
- 표준화 (Standardization)
 - 범위가 제한되지는 않지만 평균 0 표준편차 1로 바꿔줌
- 정규화 (Normalization)
 - 0에서 1사이의 값으로 변환

Part 2 - 실습

주택 가격 예측 실습 (Kaggle)

- <https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/overview>



캐글/데이콘의 파일 형태

- 메타 데이터 파일 – 데이터를 설명하는 데이터
 - 칼럼 정보
 - 카테고리 칼럼의 정보
- 실제 데이터 파일
 - 학습 데이터 (train.csv)
 - 테스트 데이터 (test.csv)
- 제출 파일 형식 – sample_submission.csv (n,2)
 - 인덱스
 - 정답 (정수/실수)

Kaggle 주택 예측 가격 구조

인덱스	파일명	뜻	형태
1	data_description.txt	메타데이터 파일	
2	sample_submission	제출 파일	
3	test.csv	테스트 데이터 (특성O /정답 O)	(1459, 80)
4	train.csv	학습 데이터 (특성O /정답 X)	(1460, 81)

리더보드 구조

- Test data의 정답에 대한 정확도로 리더보드 등재

인덱스	꽃받침 길이	꽃받침 너비	꽃잎 길이	품종
0	5.1	3.5	1.4	0
1	4.9	3.0	1.4	0
2	6.7	3.0	5.2	1
3	6.3	2.5	5.0	2
4	5.0	3.6	1.4	2
5	4.6	3.1	5.4	1
6	5.0	3.0	1.6	2
7	4.7	3.2	1.6	
8	6.7	2.5	5.0	
9	6.3	3.0	5.2	

Train X -data feature

Train y -data target

예측

정확도 측정

데이터 칼럼 분석시 참고

- 칼럼을 제공된 메타데이터와 비교하여 분석하다 보면 간혹 칼럼명이 일치하지 않는 경우가 있는데 위치가 동일하다면 보통 같다.

[메타데이터]

[실제 데이터의 변수]

Bedroom: Bedrooms above grade (does NOT include basement bedrooms)

Kitchen: Kitchens above grade

KitchenQual: Kitchen quality

Ex	Excellent
Gd	Good
TA	Typical/Average
Fa	Fair
Po	Poor

```
inSE', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath',  
, 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual',  
vGrd', 'Functional', 'Fireplaces', 'FireplaceQu', 'GarageType',  
Blt', 'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQual',
```

과제

- 데이터 전처리 끝내오기
 - > 카테고리 인코딩
 - > 결측치 처리
 - > 스케일 맞추기
 - > 각각 칼럼에 맞게
 - +어떤 알고리즘을 사용할지+그 이유는 뭔지
 - +주석으로 행위에 대한 이유
- 데이터 전처리 끝내오기

감사합니다.
