programmers

4. Regression 모델 만들기

Supervised Learning 중의 하나인 Regression 모델링에 대해 배워보자

한기용

keeyonghan@hotmail.com

programmers

목차

programmers

3장 숙제 리뷰

3장 Classification 숙제를 리뷰해보자

Classification 숙제 리뷰

- Age 필드의 값을 평균으로 채우는 일을 SimpleImputer로 해보기
- Embarked 필드의 경우 아래 해보기
  - 비워진 값의 경우 Embarked 값 중 가장 흔한 값을 사용해서 SimpleImputer로 채우기
  - Embarked를 OneHotEncoder를 사용해서 변환후 그걸 학습하는 필드 중의 하나로 추가해보기
- 숙제 리뷰

programmers

Regression 모델링이란?

Regression 모델에 대해 학습해보자

Regression 모델링 정의

- 회귀 모델은 연속적인 값을 예측하기 위해 사용되는 알고리즘
- 예: 주택 가격 예측, 주식 가격 예측 등등

Regression 알고리즘의 종류

- Linear Regression
  - Lasso Regression, Ridge Regression: Overfitting 이슈를 경감하기 위한 방법
- Polynomial Regression
- Decision Tree
- Random Forests
- Deep Learning

Regression 성능 평가

- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)

실제 판매 가격	예측 판매 가격
120	110
75	81
99	98

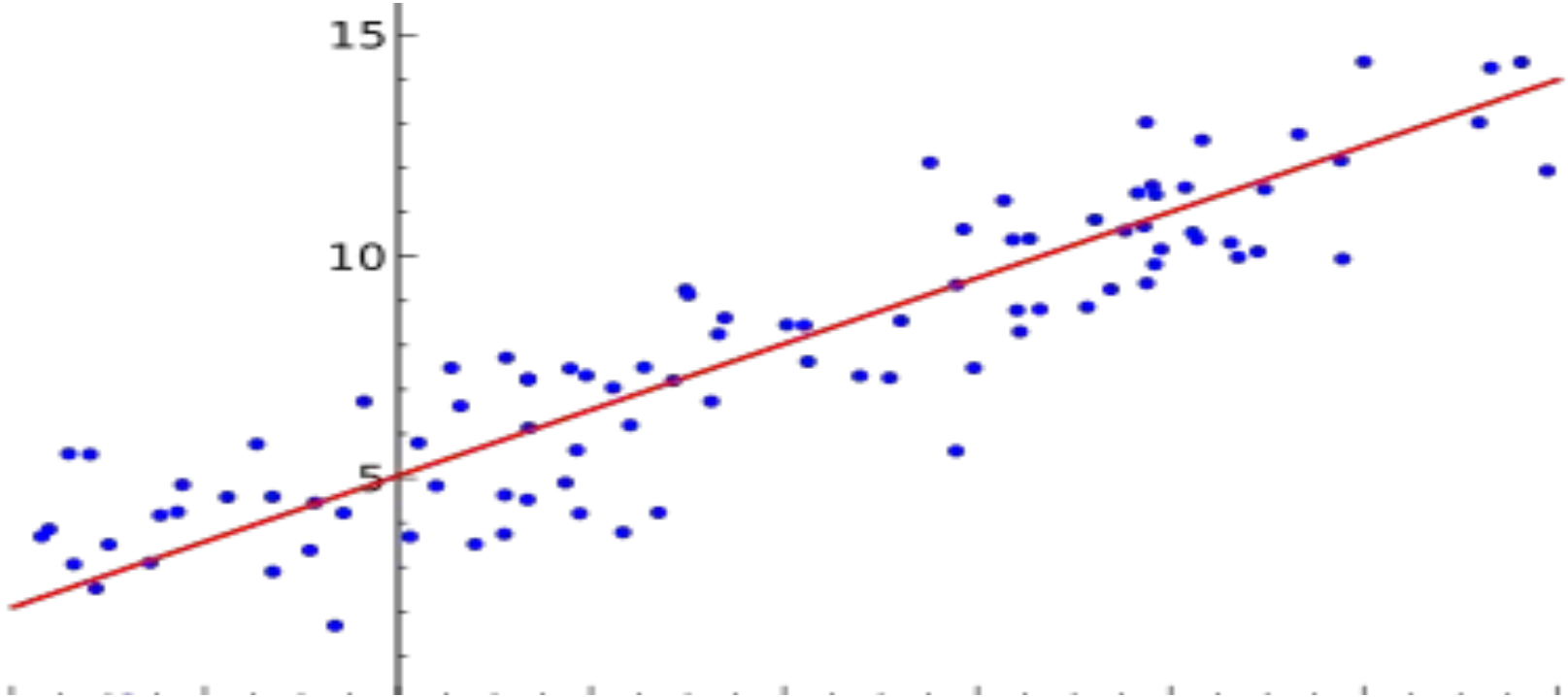
MAE	MSE	RMSE
5.67	45.67	6.75

programmers

Regression 모델 종류

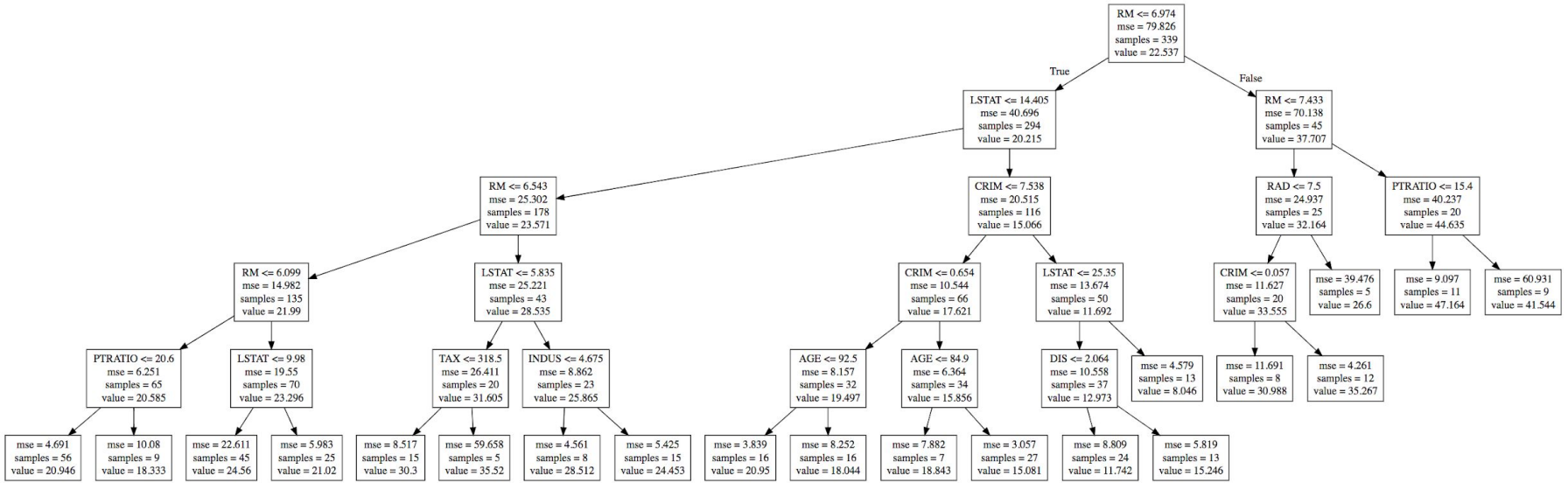
Regression 타입에 대해 알아보자

Linear Regression



Decision Tree

- Classification에도 사용 가능

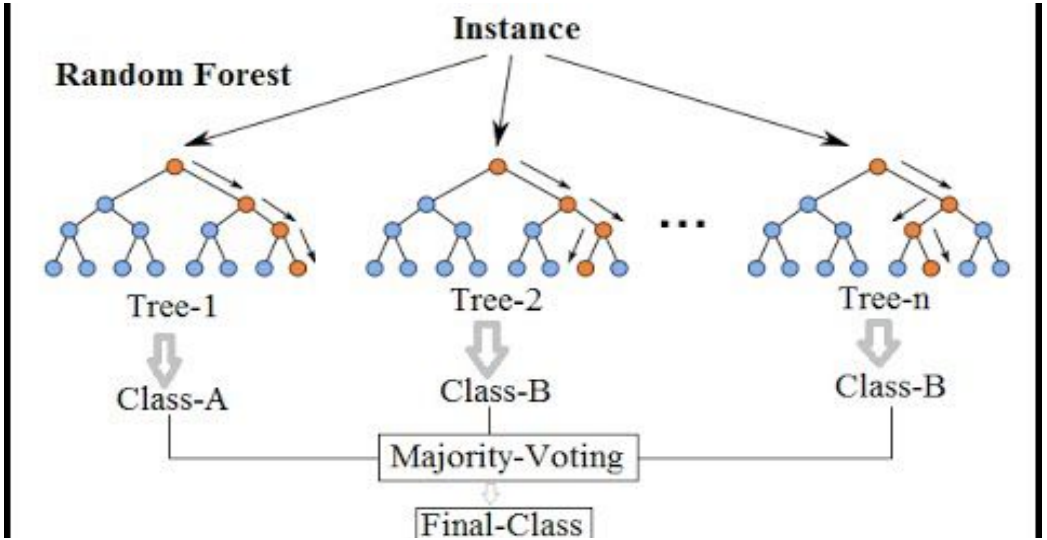


Decision Tree란?

- Classification에도 사용 가능
- 직관적으로 이해가 가능한 머신 러닝 알고리즘
- Grid Search를 통해 트리 구성이 가능
- Overfitting이 다른 알고리즘보다 쉽게 발생
- Scikit-Learn에는 두 종류의 Decision Tree
  - DecisionTreeRegressor
  - DecisionTreeClassifier

Random Forest

- Classification에도 사용 가능

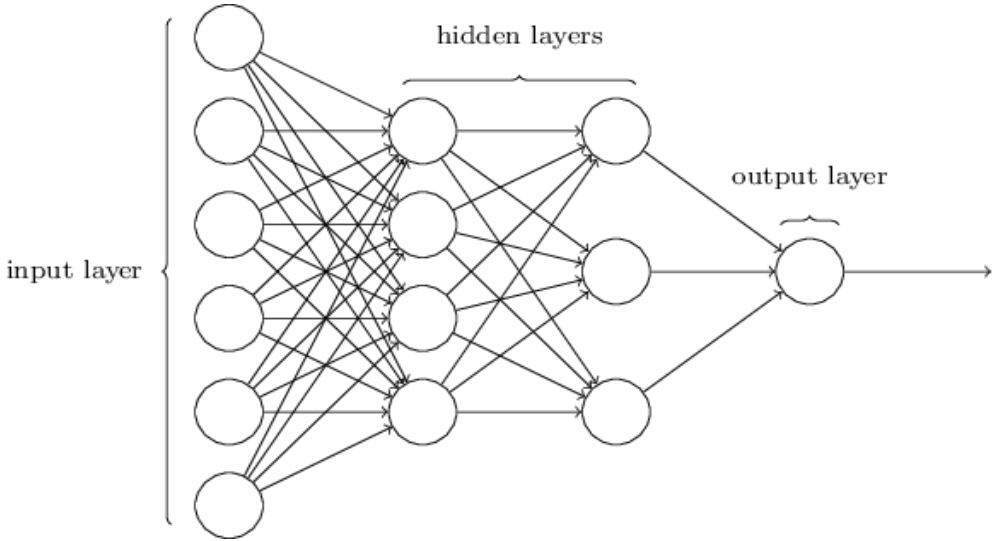


Random Forest란?

- 앙상블 학습방법의 일종
- Decision Tree의 Overfitting 이슈를 막기 위해 다수의 decision tree를 사용하여 그 결과를 평균 (혹은 다수결)하는 방식
- Scikit-Learn에는 RandomForestClassifier 클래스 존재

Deep Learning

- Classification에도 사용 가능



programmers

Regression 모델 실습 #1

주택 가격 예측 Regression 모델을 만들어보자

보스턴 주택가격 예측

- 1970년대 미국 인구조사 서비스 (US Census Service)에서 보스턴 지역의 주택 가격 데이터를 수집한 데이터를 기반으로 모델 빌딩
- 트레이닝셋은 여기서 다운로드 (원래 데이터는 CMU Stat Lib Archive)
  - 개별 주택가격의 예측이 아니라 지역별 중간 주택가격 예측임
- Regression 알고리즘 사용 예정
  - 연속적인 주택가격을 예측이기에 Classification 알고리즘은 사용불가

보스턴 주택가격 트레이닝 셋 보기

- 총 506개의 레코드로 구성되며 13개의 피쳐와 레이블 필드(주택가격)로 구성
  - 506개 동네의 주택 중간값 데이터임 (개별 주택이 아님에 유의)
  - 14번째 필드가 바로 예측해야하는 중간 주택 가격

필드 이름	설명
1. CRIM	주택이 있는 지역의 인당 범죄율
2. ZN	25000 sqft (대략 700평) 이상의 땅이 주거지역으로 설정된 비율
3. INDUS	에이커당 공업단지의 비율
4. CHAS	주택이 강가에 위치한 비율
5. NOX	산화질소 농도로 오염정도를 나타냄
6. RM	주택당 평균 방의 수
7. AGE	1940년 전에 지어진 주택의 비율

필드 이름	설명
8. DIS	보스턴 지역 고용 센터까지의 평균 거리
9. RAD	고속도로 접근성에 대한 인덱스
10. TAX	재산세 (주택가격 $10K 기준)
11. PTRATO	초등학교 학생-선생님의 비율
12. B	흑인 인구의 비율
13. LSTAT	저소득자의 인구 비율
14. MEDV	**천불 단위의 주택 평균값**

실습

1. Linear Regression 사용
2. 구글 Colab 상에서 실습 진행

programmers

Regression 모델 실습 #2

보스턴 주택 가격 Regression 모델을 Decision Tree로 만들어보자

Grid Search란 무엇인가?

- 주어진 모델의 최적의 하이퍼파라미터를 찾기 위해 사용
- 하이퍼파라미터 조합(그리드)을 시스템적으로 탐색하여 최적의 조합 탐색

```
parameters = {
   'max_depth':(1,2,3,4,5,6,7,8,9,10),  # 트리의 최대 깊이. 트리가 너무 깊어지면 과적합(overfitting)이 발생
   'min_samples_split': [2, 10, 20], # 노드를 분할하기 위해 필요한 최소 샘플 수.
   'min_samples_leaf': [1, 5, 10],  # 리프 노드가 되기 위해 필요한 최소 샘플 수
   'max_leaf_nodes': [5, 10, 20]  # 트리가 가질 수 있는 리프 노드의 최대 개수
}
```

- 위 파라미터들은 트리가 얼마나 깊게, 얼마나 세분화되어 성장할지를 결정
    - 노드가 너무 많아지면 과적합 발생. 너무 적으면 성능 저하. 적당한 수가 필요

Grid Search 수행 과정

```
from sklearn.model_selection import GridSearchCV
from sklearn.tree import DecisionTreeRegressor

parameters = {
    'max_depth':(1,2,3,4,5,6,7,8,9,10),
    'min_samples_split': [2, 10, 20],
    'min_samples_leaf': [1, 5, 10],
    'max_leaf_nodes': [5, 10, 20],
}
regressor = DecisionTreeRegressor()
regressors = GridSearchCV(regressor, parameters, cv=5, scoring='neg_mean_squared_error')
regressors.fit(X_train, Y_train)
reg = regressors.best_estimator_
```

실습

1. Decision Tree 사용
2. 구글 Colab 상에서 실습 진행

programmers

숙제

이번 챕터 숙제에 대해 알아보자

숙제

Regression 모델을 Random Forest로 만들어보기

- 앞서 보스턴 주택 가격 예측 Regression 모델을 Random Forest로 만들어보기

```
from sklearn.ensemble import RandomForestRegressor


…
rf = RandomForestRegressor(
    n_estimators=100,   # 트리의 갯수로 디폴트 값도 100
    random_state=42
)
rf.fit(X_train, y_train)
```

programmers

Q & A

오늘 강의에 대해서 궁금한 부분이 있으면 알려주세요!