ch02 앙상블 기법- randomForest(3) - 집값 예측

학습 내용

01. 집값 예측 기본 모델 개선하기

캐글 코리아 2차 대회 데이터 셋 데이터

- https://www.kaggle.com/c/2019-2nd-ml-month-with-kakr/data

컬럼명	의미	값(기타)
ID	집을 구분하는 번호	
date	집을 구매한 날짜	
price	집의 가격(Target variable)	
bedrooms	침실의 수	
bathrooms	화장실의 수	
sqft_living	주거 공간의 평방 피트(면적)	
sqft_lot	부지의 평방 피트(면적)	
floors	집의 층 수	
waterfront	집의 전방에 강이 흐르는지 유무 (a.k.a. 리버뷰)	
view	집이 얼마나 좋아 보이는지의 정도	
condition	집의 전반적인 상태	
grade	King County grading 시스템 기준으로 매긴 집의 등급	
sqft_above	지하실을 제외한 평방 피트(면적)	
sqft_basement	지하실의 평방 피트(면적)	
yr_built	지어진 년도	
yr_renovated	집을 재건축한 년도	
zipcode	우편번호	
lat	위도	
long	경도	
sqft_living15	2015년 기준 주거 공간의 평방 피트(면적, 집을 재건축했다면, 변화가 있을 수 있음)	
sqft_lot15	2015년 기준 부지의 평방 피트(면적, 집을 재건축했다면, 변화가 있을 수 있음)	

In [ ]:
```
import pandas as pd

train = pd.read_csv("house_train.csv")
test = pd.read_csv("house_test.csv")
```

In [ ]:
```
train.columns
```

Out[ ]:
```
Index(['id', 'date', 'price', 'bedrooms', 'bathrooms', 'sqft_living',
       'sqft_lot', 'floors', 'waterfront', 'view', 'condition', 'grade',
       'sqft_above', 'sqft_basement', 'yr_built', 'yr_renovated', 'zipcode',
       'lat', 'long', 'sqft_living15', 'sqft_lot15'],
      dtype='object')
```

- 예측하고자 하는 값(target)이 price

In [ ]:
```
X_all = train.drop(['price'], axis=1)
y = train['price']

print(type(X_all), type(y), X_all.shape, y.shape)
```

```
<class 'pandas.core.frame.DataFrame'> <class 'pandas.core.series.Series'> (15035, 20) (15035,)
```

In [ ]:
```
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
import matplotlib.pyplot as plt
import numpy as np
```

In [ ]:
```
sel = ['sqft_living', 'sqft_lot', 'bedrooms']  # 'bedrooms' , 'bathrooms',
X = X_all[sel]
y = train['price']

nor_X = MinMaxScaler().fit_transform(X)  # 입력 데이터 정규화
print("정규화 : ", nor_X.shape, y.shape)

#  정규화 데이터 사용
X_train, X_test, y_train, y_test = train_test_split(nor_X, y,
                                                    random_state=42)


# 정규화 데이터 사용 안함.
# X_train, X_test, y_train, y_test = train_test_split(X, y,
#                                                     random_state=42)
```

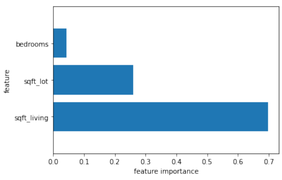
```
정규화 :  (15035, 3) (15035,)
```

In [ ]:
```
model = RandomForestRegressor(n_estimators=5, random_state=2)  # 5개의 트리
print( model.fit(X_train, y_train) )
print( model.score(X_train, y_train))
print( model.score(X_test, y_test))
```

```
RandomForestRegressor(n_estimators=5, random_state=2)
0.8922137121180739
0.3793764028830927
```

In [ ]:
```
# model : 모델
# n_features : feature(변수의 개수)
# feature_names : 특성의 이름
def plot_feature_important_up(model, n_features, feature_names):
  imp = model.feature_importances_     # feature의 중요도
  plt.barh(range(n_features) , imp, align='center')  # 그래프(가로 막대 그래프)
  plt.yticks(np.arange(n_features), feature_names)  #y축의 축의 값
  plt.xlabel("feature importance")   # x축 레이블(제목)
  plt.ylabel("feature")              # y축 제목
  plt.ylim(-1, n_features)           # y축의 범위 지정
```

In [ ]:
```
feature_names = sel   #  선택된 피처의 이름
n_features = X.shape[1]  # 선택된 피처의 개수
plot_feature_important_up(model, n_features, feature_names) # 피처의 중요도 확인
```



실습

- 주어진 여러가지 특성을 넣어 성능을 개선시켜보자.

In [ ]:
```
sel = ['id', 'bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot',
       'floors', 'waterfront', 'view', 'condition', 'grade', 'sqft_above',
       'sqft_basement', 'yr_built', 'yr_renovated', 'zipcode', 'lat', 'long', 'sqft_living15', 'sqft_lot15']


X = X_all[sel]
y = train['price']

nor_X = MinMaxScaler().fit_transform(X)  # 입력 데이터 정규화
print("정규화 : ", nor_X.shape, y.shape)

#  정규화 데이터 사용
X_train, X_test, y_train, y_test = train_test_split(nor_X, y,
                                                    random_state=42)


# 정규화 데이터 사용 안함.
# X_train, X_test, y_train, y_test = train_test_split(X, y,
#                                                     random_state=42)
```

```
정규화 :  (15035, 19) (15035,)
```

In [ ]:
```
model = RandomForestRegressor(random_state=2)  # 5개의 트리
print( model.fit(X_train, y_train) )
print( model.score(X_train, y_train))
print( model.score(X_test, y_test))
```

```
RandomForestRegressor(random_state=2)
0.9827015345599711
0.8350532950678641
```

결과

```
sel = ['sqft_living', 'bathrooms', 'grade', 'sqft_above', 'sqft_lot15']
결정계수 =  학습용: 0.848818, 테스트용: 0.472017

sel = ['bedrooms', 'bathrooms', 'grade', 'sqft_living', 'sqft_lot']
결정계수 : 학습용 : 0.94803, 테스트용 : 0.56981

 * 훈련 세트 정확도 : 0.950, 테스트 세트 정확도 : 0.565
 * 기본 : 주거공간의 면적, 부지의 면적, 집의 층수,
 추가 변수 : 강이 흐르는지 유무(waterfront), 지어진 년도(yr_built)
 n_estimators : 100

 * 훈련 세트 정확도 : 0.948, 테스트 세트 정확도 : 0.584
 * 기본 : 주거공간의 면적, 부지의 면적, 집의 층수,
 추가 변수 : 지어진년도(yr_built), 집이 얼마나 좋아 보이는지의 정도(view)
 n_estimators : 100

sel = ['id', 'bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot',
       'floors', 'waterfront', 'view', 'condition', 'grade', 'sqft_above', 'sqft_basement', 'yr_built',
'yr_renovated', 'zipcode', 'lat', 'long', 'sqft_living15', 'sqft_lot15']
결정계수 : 학습용 : 0.98126, 테스트용 : 0.8343

sel = ['id', 'bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot',
       'floors', 'waterfront', 'view', 'condition', 'grade', 'sqft_above','sqft_basement', 'yr_built',
'yr_renovated', 'zipcode', 'lat', 'long' ]
결정계수 : 학습용 : 0.98177, 테스트용 : 0.84573

 * 훈련 세트 정확도 : 0.9816, 테스트 세트 정확도 : 0.8485
 기본 : 주거공간의 면적, 부지의 면적, 집의 층수,
 추가 : 'bathrooms', 'floors', 'condition', 'sqft_above', 'sqft_basement',
    'view', 'grade', 'yr_built', 'yr_renovated', 'lat', 'long', 'waterfront']
 n_estimators : 100
```

실습해 보기 3

```
데이터 셋 : 유방암 데이터 셋
from sklearn.datasets import load_breast_cancer

위의 데이터 셋을 이용하여 모델을 만들어보자.
(1) 랜덤 포레스트를 이용하여 훈련 세트 정확도, 테스트 세트 정확도를 확인해 보자.
(랜덤 포레스트 트리의 개수 = 5개, random_state=0, 최대 변수 선택 = 4)
```

도전 실습

- 2차 캐글 대회의 데이터 셋을 활용하여 랜덤 포레스트 모델을 구해보자.
- url : https://www.kaggle.com/c/2019-2nd-ml-month-with-kakr의 대회
- 여러가지 변수 파라미터를 조절하여 RMSE가 가장 좋을때가 언제인가?

History

- 2020/12 업데이트 v11
- Machine Learning with sklearn @ DJ,Lim