04.Valuation tool

April 30, 2021

[25]:
```
from sklearn.datasets import load_boston
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

import pandas as pd
import numpy as np
```

[26]:
```
# Gather Data
boston_dataset = load_boston()
data = pd.DataFrame(data=boston_dataset.data,columns=boston_dataset.
 ↪feature_names)
features = data.drop(['INDUS','AGE'],axis=1)

log_prices = np.log(boston_dataset.target)
target = pd.DataFrame(log_prices,columns=['PRICE'])
```

[27]:
```
CRIME_IDX =0
ZN_IDX =1
CHAS_IDX =2
RM_IDX=4
PTRATIO_IDX=8

# property_stats = np.ndarray(shape=(1,11))
# property_stats[0][CRIME_IDX]=features['CRIM'].mean()
# property_stats[0][ZN_IDX]=features['ZN'].mean()
# property_stats[0][CHAS_IDX]=features['CHAS'].mean()

property_stats = features.mean().values.reshape(1,11)
```

[28]:
```
regr = LinearRegression().fit(features,target)
fitted_vals=regr.predict(features)

#Challenge: Calculate MSE and RMSE using sklearn
MSE = mean_squared_error(target,fitted_vals)
RMSE = np.sqrt(MSE)
```

[29]:
```
def␣
↪get_log_estimate(nr_room,students_per_classroom,next_to_river=False,high_confidence=True):
↪

    # Configure property
    property_stats[0][RM_IDX]=nr_room
    property_stats[0][PTRATIO_IDX]=students_per_classroom
    if(next_to_river):
        property_stats[0][CHAS_IDX]=1
    else:
        property_stats[0][CHAS_IDX]=0

    # Make Prediction
    log_estimate = regr.predict(property_stats)[0][0]

    # Calc Range
    if(high_confidence):
        upper_bound=log_estimate+ 2*RMSE
        lower_bound=log_estimate-2*RMSE
        interval=95

    else:
        upper_bound=log_estimate+ RMSE
        lower_bound=log_estimate- RMSE
        interval=68


    return log_estimate,upper_bound,lower_bound,interval
```

[30]:
```
def get_dollar_estimate(rm,ptratio,chas=False,large_range=True):
    #Challenge: Writethe python code to convertlog price using 1970s prices as␣
↪well as upper and lower bounds to today's prices Round the value to 1000␣
↪dollars
    """Estimate the price of a property in Boston.

    Keyword arguments:
    rm -- number of rooms in the property.
    ptratio -- number of students per teacher in the classroom for the school␣
↪in area.
    chas -- True if the property is next to the river, False otherwise.
    large_range -- True for a 95% prediction interval, False for 68% interval.
    """
    if rm < 1 or ptratio < 1:
        print('This is unrealistic. Try Again')
        return
    ZILLOW_MEDIAN_PRICE = 583.3
    SCALE_FACTOR = ZILLOW_MEDIAN_PRICE/np.median(boston_dataset.target)
```

```
    log_est,upper,lower,conf =␣
↪get_log_estimate(rm,students_per_classroom=ptratio,next_to_river=chas,high_confidence=large

    # convert today's dollar
    dollar_est = (np.e**log_est)*1000*SCALE_FACTOR
    dollar_hi = (np.e**upper)*1000*SCALE_FACTOR
    dollar_low = (np.e**lower)*1000*SCALE_FACTOR

    #round the dollar to nearest thousand
    round_est= np.around(dollar_est,-3)
    round_hi= np.around(dollar_hi,-3)
    round_low= np.around(dollar_low,-3)

    print(f'The estimated property value is {round_est}. ')
    print(f'At {conf}% confidence the valuation range is')
    print(f'USD {round_low} at the lower end to {round_hi} at the higher end. ')
```

[31]:
```
get_dollar_estimate(rm=2,ptratio=30,chas=True)
```

```
The estimated property value is 278000.0.
At 95% confidence the valuation range is
USD 191000.0 at the lower end to 404000.0 at the higher end.
```

[32]:
```
import boston_valuation as val
```

[33]:
```
val.get_dollar_estimate(6,12,True)
```

```
The estimated property value is 783000.0.
At 95% confidence the valuation range is
USD 538000.0 at the lower end to 1139000.0 at the higher end.
```

[34]:
```
val.get_dollar_estimate(8,15,False)
```

```
The estimated property value is 755000.0.
At 95% confidence the valuation range is
USD 519000.0 at the lower end to 1099000.0 at the higher end.
```

[ ]:
```

```