

Bootstrap_assignment

December 23, 2020

1 Bootstrap assignment

There will be some functions that start with the word “grader” ex: grader_sampples(), grader_30().. etc, you should not change those function definition.

Every Grader function has to return True.

Importing packages

```
[1]: import numpy as np # importing numpy for numerical computation
from sklearn.datasets import load_boston # here we are using sklearn's boston
    ↳dataset
from sklearn.metrics import mean_squared_error # importing mean_squared_error
    ↳metric
```

```
[2]: boston = load_boston()
x=boston.data #independent variables
y=boston.target #target variable
```

```
[3]: x.shape
```

```
[3]: (506, 13)
```

```
[69]: x[:5]
```

```
[69]: array([[6.3200e-03, 1.8000e+01, 2.3100e+00, 0.0000e+00, 5.3800e-01,
        6.5750e+00, 6.5200e+01, 4.0900e+00, 1.0000e+00, 2.9600e+02,
        1.5300e+01, 3.9690e+02, 4.9800e+00],
        [2.7310e-02, 0.0000e+00, 7.0700e+00, 0.0000e+00, 4.6900e-01,
        6.4210e+00, 7.8900e+01, 4.9671e+00, 2.0000e+00, 2.4200e+02,
        1.7800e+01, 3.9690e+02, 9.1400e+00],
        [2.7290e-02, 0.0000e+00, 7.0700e+00, 0.0000e+00, 4.6900e-01,
        7.1850e+00, 6.1100e+01, 4.9671e+00, 2.0000e+00, 2.4200e+02,
        1.7800e+01, 3.9283e+02, 4.0300e+00],
        [3.2370e-02, 0.0000e+00, 2.1800e+00, 0.0000e+00, 4.5800e-01,
        6.9980e+00, 4.5800e+01, 6.0622e+00, 3.0000e+00, 2.2200e+02,
        1.8700e+01, 3.9463e+02, 2.9400e+00],
        [6.9050e-02, 0.0000e+00, 2.1800e+00, 0.0000e+00, 4.5800e-01,
        7.1470e+00, 5.4200e+01, 6.0622e+00, 3.0000e+00, 2.2200e+02,
        1.8700e+01, 3.9690e+02, 5.3300e+00]])
```

1.1 Task 1

Step - 1

- Creating samples Randomly create 30 samples from the whole boston data points
 - Creating each sample: Consider any random 303(60% of 506) data points from whole data set and then replicate any 203 points from the sampled points

For better understanding of this procedure let's check this example, assume we have 10 data points [1,2,3,4,5,6,7,8,9,10], first we take 6 data points randomly, consider we have selected [4, 5, 7, 8, 9, 3] now we will replicate 4 points from [4, 5, 7, 8, 9, 3], consider they are [5, 8, 3, 7] so our final sample will be [4, 5, 7, 8, 9, 3, 5, 8, 3, 7]

- Create 30 samples
 - Note that as a part of the Bagging when you are taking the random samples make sure each of the sample will have different set of columns Ex: Assume we have 10 columns [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] for the first sample we will select [3, 4, 5, 9, 1, 2] and for the second sample [7, 9, 1, 4, 5, 6, 2] and so on... Make sure each sample will have at least 3 features/columns/attributes

Step - 2

Building High Variance Models on each of the sample and finding train MSE value

- Build a regression trees on each of 30 samples.
- Compute the predicted values of each data point (506 data points) in your corpus.
- Predicted house price of i^{th} data point $y_{pred}^i = \frac{1}{30} \sum_{k=1}^{30} (\text{predicted value of } x^i \text{ with } k^{th} \text{ model})$
- Now calculate the $MSE = \frac{1}{506} \sum_{i=1}^{506} (y^i - y_{pred}^i)^2$

Step - 3

- Calculating the OOB score
- Predicted house price of i^{th} data point $y_{pred}^i = \frac{1}{k} \sum_{k=\text{model which was built on samples not included } x^i} (\text{predicted value of } x^i)$
- Now calculate the $OOBScore = \frac{1}{506} \sum_{i=1}^{506} (y^i - y_{pred}^i)^2$.

2 Task 2

- Computing CI of OOB Score and Train MSE
- Repeat Task 1 for 35 times, and for each iteration store the Train MSE and OOB score
After this we will have 35 Train MSE values and 35 OOB scores
using these 35 values (assume like a sample) find the confidence intervals of MSE and OOB Score
you need to report CI of MSE and CI of OOB Score
Note: Refer the Central_Limit_theorem.ipynb to check how to find the confidence interval

3 Task 3

- Given a single query point predict the price of house.

Consider $x_q = [0.18, 20.0, 5.00, 0.0, 0.421, 5.60, 72.2, 7.95, 7.0, 30.0, 19.1, 372.13, 18.60]$ Predict the house price for this point as mentioned in the step 2 of Task 1.