|  |  |  |  |
| --- | --- | --- | --- |
| **Data.describe()** | **checking the distributed of price variable** | **Data.shape** | **Data.head** |
| | **enginesize** | **carlength** | **carwidth** | **carheight** | **price** | | --- | --- | --- | --- | --- | | **count** | 99.000000 | 99.000000 | 99.000000 | 99.000000 | 99.000000 | | **mean** | 148.555556 | 157.334343 | 82.125253 | 63.203030 | 30965.717172 | | **std** | 23.886305 | 29.333543 | 7.765207 | 7.110519 | 16727.613490 | | **min** | 109.000000 | 115.400000 | 64.100000 | 48.800000 | 13495.000000 | | **25%** | 135.500000 | 130.400000 | 77.800000 | 57.900000 | 19116.500000 | | **50%** | 145.000000 | 168.800000 | 80.300000 | 60.800000 | 20000.000000 | | **75%** | 168.000000 | 177.500000 | 88.800000 | 67.400000 | 35987.000000 | | **max** | 190.000000 | 196.400000 | 95.300000 | 77.800000 | 66786.000000 | | 20000 13  25876 12  66786 7  56789 7  33786 7  15688 7  47342 6  35987 6  19678 6  16895 6  19897 6  18555 4  17450 4  60123 4  16500 2  13950 1  13495 1 | (99, 5) | | **enginesize** | **carlength** | **carwidth** | **carheight** | **price** | | --- | --- | --- | --- | --- | | **0** | 130 | 168.8 | 64.1 | 48.8 | 13495 | | **1** | 130 | 168.8 | 64.1 | 48.8 | 16500 | | **2** | 152 | 171.2 | 65.5 | 52.4 | 16500 | | **3** | 109 | 176.6 | 66.2 | 54.3 | 13950 | | **4** | 136 | 176.6 | 66.4 | 54.3 | 17450 | |
|  | **data\_main.corr()['price']** | **checking the missing values** | **Data.info** |
|  | enginesize 0.176891  carlength 0.052969  carwidth 0.277050  carheight 0.944604  price 1.000000  Name: price, dtype: float6 | enginesize 0  carlength 0  carwidth 0  carheight 0  price 0  dtype: int64 | bound method DataFrame.info of enginesize carlength carwidth carheight price  0 130 168.8 64.1 48.8 13495  1 130 168.8 64.1 48.8 16500  2 152 171.2 65.5 52.4 16500  3 109 176.6 66.2 54.3 13950  4 136 176.6 66.4 54.3 17450  .. ... ... ... ... ...  94 134 196.4 69.8 75.6 66786  95 135 196.4 69.8 77.8 66786  96 178 196.4 69.8 77.8 66786  97 169 145.7 69.8 77.8 66786  98 169 145.7 69.8 77.8 66786  [99 rows x 5 columns]> |
| **Improving accuracy using gradient boosting model** | **Accuracy using linear regression model** | **Splitting data into data test and data train** | **data['price'].value\_counts()** |
| Train R2-score: 0.99  Test R2-score: 0.99  Train CV score: [0.98307736 0.99517648 0.9088012 0.98358293 0.79234892]  Train CV mean: 0.93 | Train R2-score: 0.91  Test R2-score: 0.95  Train CV score: [0.89008164 0.9260122 0.8557138 0.93266952 0.75807153]  Train CV mean: 0.87 | x train: (79, 3)  x test: (8, 3)  y train: (79,)  y test: (8,) | 20000 13  25876 12  66786 7  56789 7  33786 7  15688 7  47342 6  35987 6  19678 6  16895 6  19897 6  18555 4  17450 4  60123 4  16500 2  13950 1  13495 1  Name: price, dtype: int64 |
|  |  |  | **Compression the accuracy using two models**   | **Model** | **R Squared(Train)** | **R Squered(Test)** | **Cv score mean(Train)** | | --- | --- | --- | --- | | **0** | LinearRegression | 0.91 | 0.95 | 0.87 | | **1** | GradientBoosting | 0.99 | 0.99 | 0.93 | |