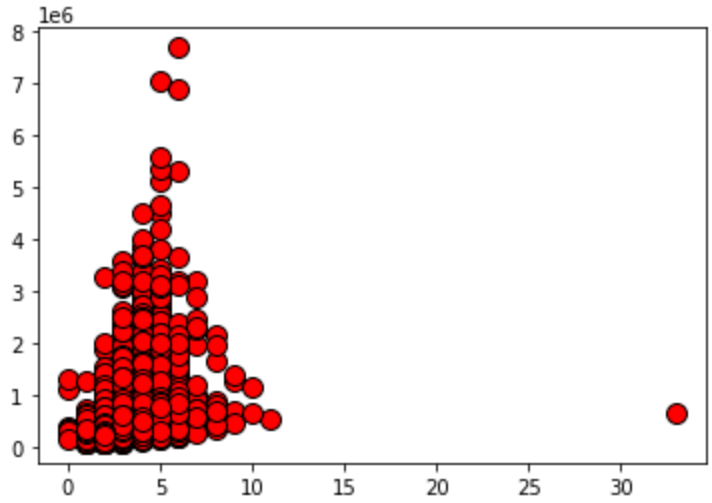
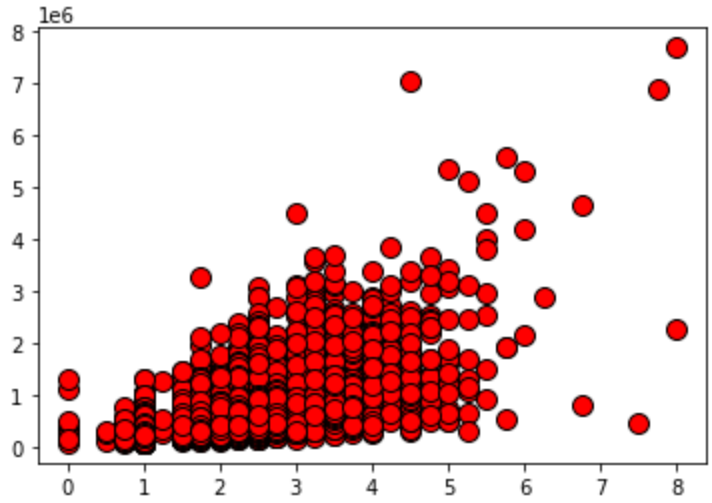
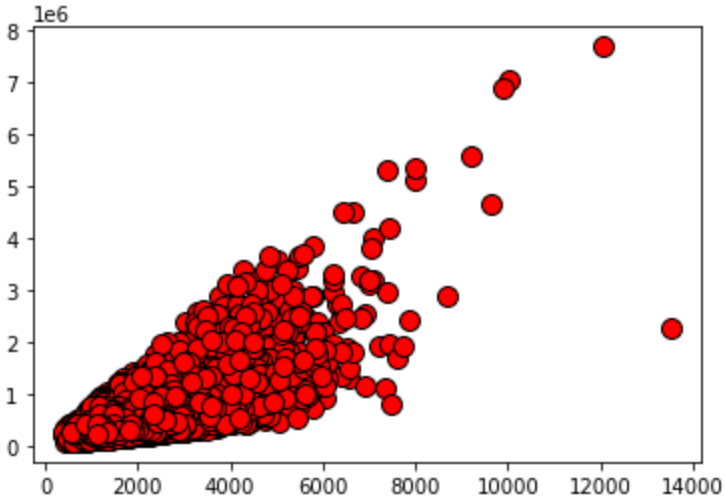
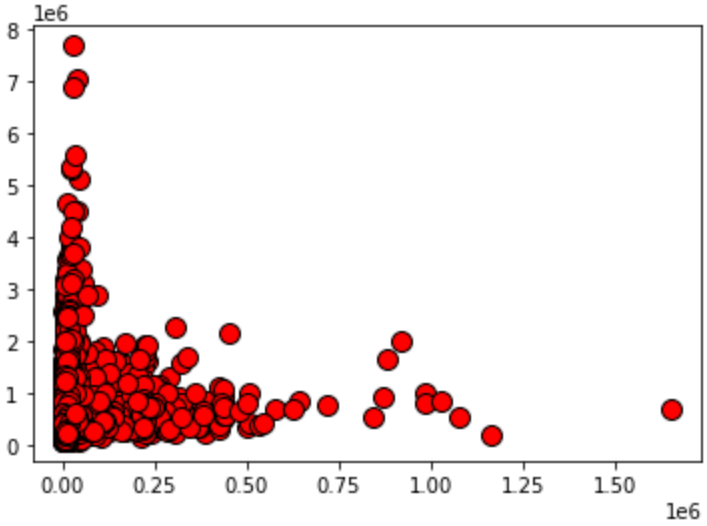
Report 1 Machine Learning

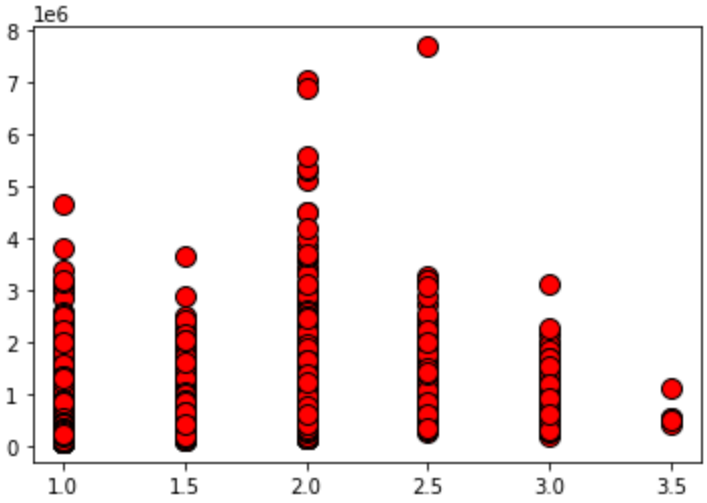
Plot each feature with y(prices) in their respective order:

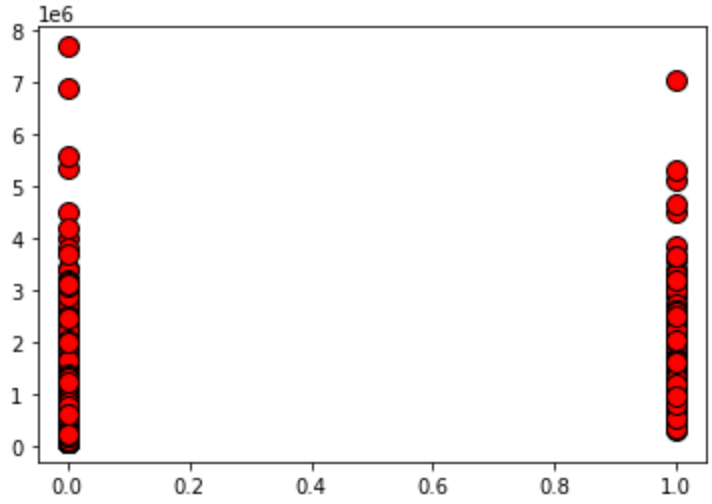


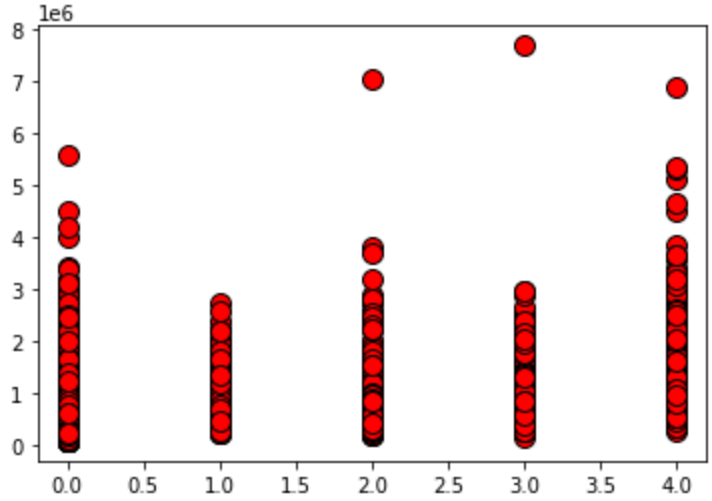


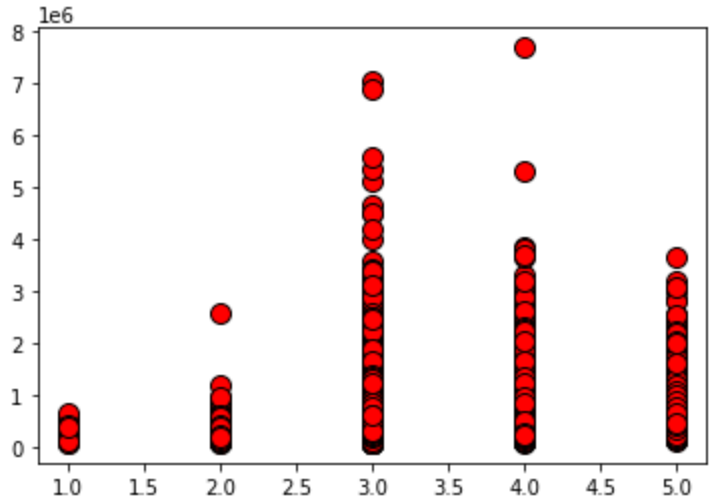

1e6
8
7
6
5
4
3
2
1
0
0
2000
4000
6000
8000
10000
12000
14000

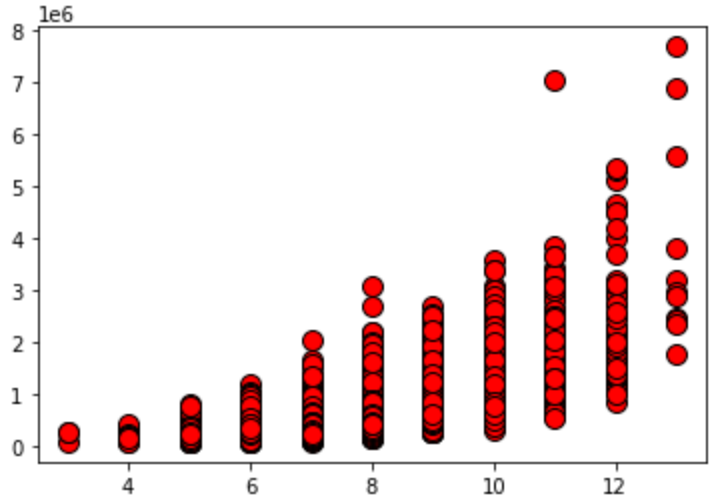

1e6
8
7
6
5
4
3
2
1
0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1e6

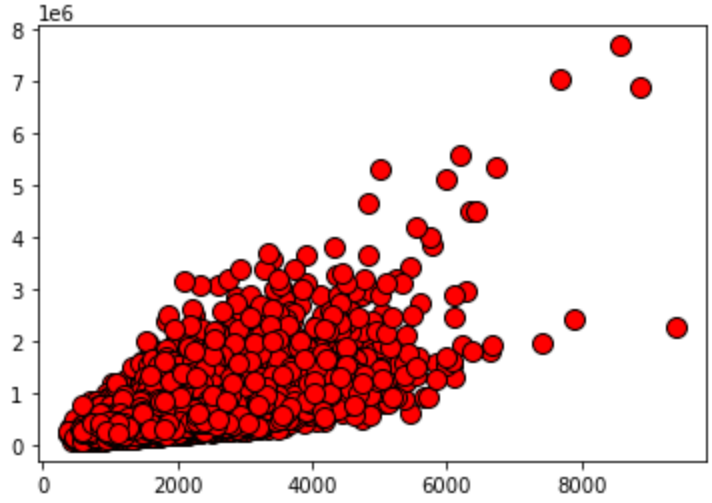

1e6
8
7
6
5
4
3
2
1
0
1.0
1.5
2.0
2.5
3.0
3.5

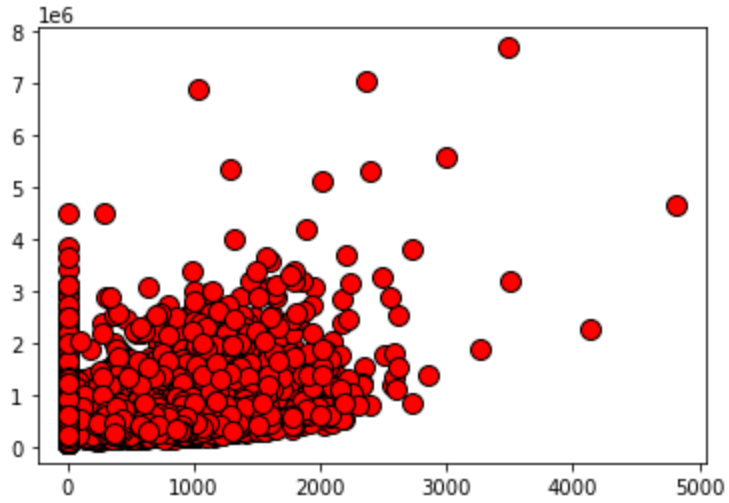

1e6
8
7
6
5
4
3
2
1
0
0.0
0.2
0.4
0.6
0.8
1.0

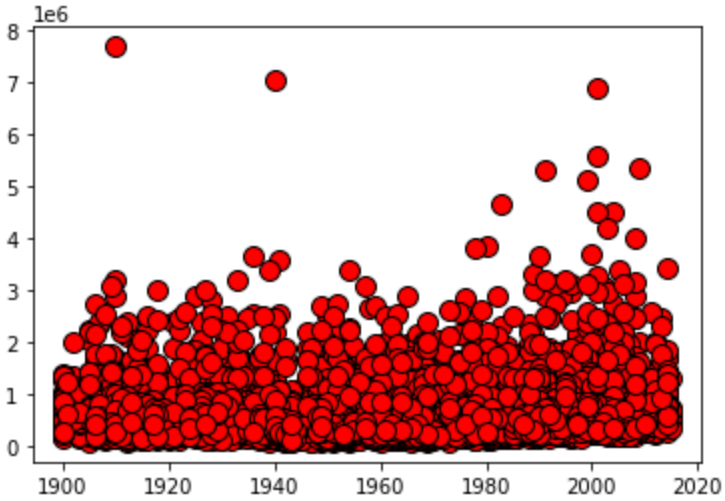

1e6
8
7
6
5
4
3
2
1
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

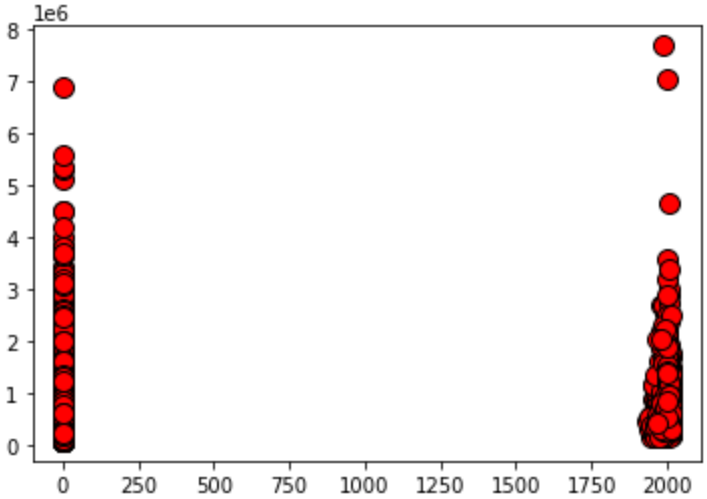

1e6
8
7
6
5
4
3
2
1
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

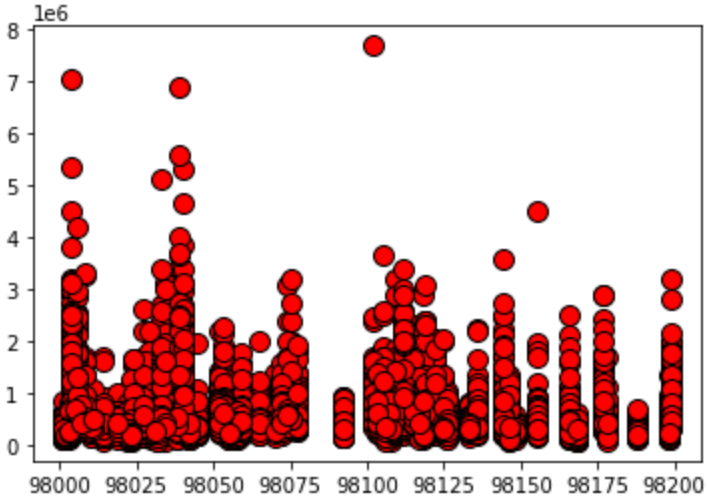

1e6
8
7
6
5
4
3
2
1
0
4
6
8
10
12

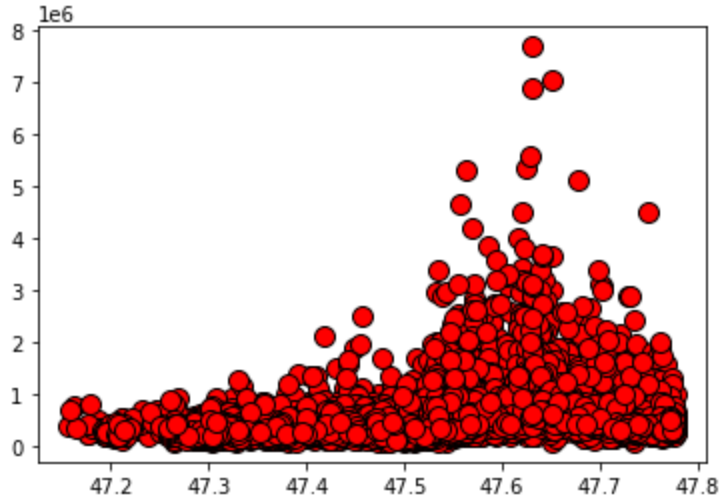

1e6
8
7
6
5
4
3
2
1
0
0
2000
4000
6000
8000

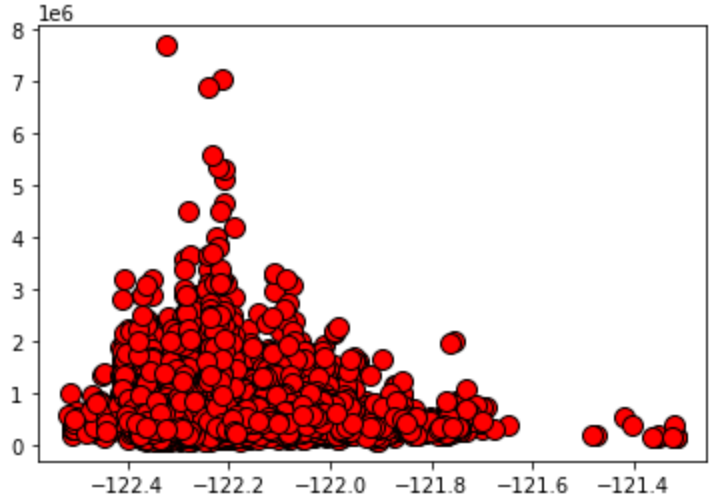

1e6
8
7
6
5
4
3
2
1
0
0
1000
2000
3000
4000
5000

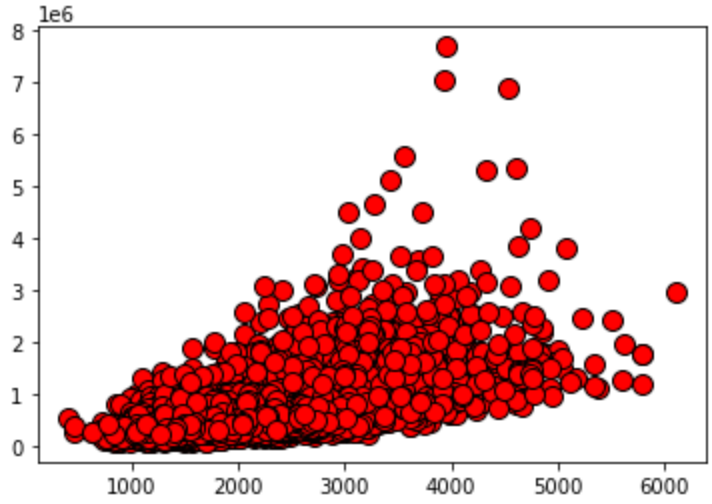

1e6
8
7
6
5
4
3
2
1
0
1900
1920
1940
1960
1980
2000
2020

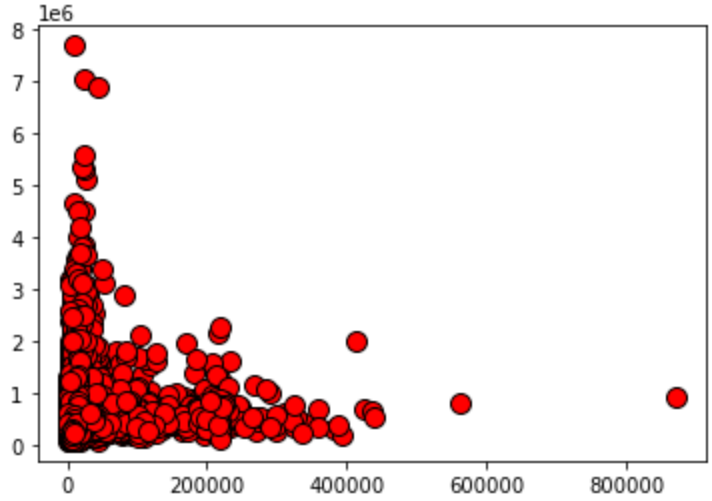

1e6
8
7
6
5
4
3
2
1
0
0
250
500
750
1000
1250
1500
1750
2000


1e6
8
7
6
5
4
3
2
1
0
98000
98025
98050
98075
98100
98125
98150
98175
98200


1e6
8
7
6
5
4
3
2
1
0
47.2
47.3
47.4
47.5
47.6
47.7
47.8


1e6
8
7
6
5
4
3
2
1
0
−122.4
−122.2
−122.0
−121.8
−121.6
−121.4





Conclusion: the 12th and 14th features can be excluded as they have scattered graphs

Model Selection:

1. **Deleting features:**
   Cost Function using Gradient Descent before deleting any features:
   19869213160.99932
   The cost function increases when removing any feature except for the 5th feature.
   Cost Function using Gradient Descent after deleting the 5th feature (floors):
   19868580710.408558
   There is no much difference, so I will not delete any feature

2. **Divide the data into training (60%), Cross Validation (cv) (20%), and testing (20%)**

3. **Choose the hypothesis polynomial which has the lowest cross-validation error**
    - We tried 5 hypotheses (1st order polynomials -> 5th order polynomials)

      21081976728.368187 train error 1
      18731658750.152397 cv error 1
      #########

      17045994045.187153 train error 2
      20314324271.591145 cv error 2
      #########

      16163167507.723595 train error 3
      22398862689.870476 cv error 3
      #########

      16028237617.19586 train error 4
      22129791856.004166 cv error 4
      #########

      15950797770.978123 train error 5
      21849196333.432262 cv error 5
      #########

      Hypothesis 1 has the lowest cv error so I chose it and will find its testing error
      17796063000.07086 test error 1

4. **Use K-Fold Sampling to partition the data and compute average cost of each test set**
   20040609743.29821 average error of 1st hyp Kfolding
   18206670493.151264 average error of 2nd hyp Kfolding
   18863779607.37979 average error of 3rd hyp Kfolding
   18901274716.42986 average error of 4th hyp Kfolding
   1.0009293586734119e+42 average error of 5th hyp Kfolding

   Hypothesis 2 has the lowest average testing error

<u>Regularization:</u>

1. **Compute regularized cost function**
2. **Compute regularized gradient descent**

21075532967.92147 train error regularized 1
18796526751.00079 cv error regularized 1
#########

16238724157.916054 train error regularized 2
22028557503.034565 cv error regularized 2
#########

15783911963.79675 train error regularized 3
22501437881.914845 cv error regularized 3
#########

15658628628.743156 train error regularized 4
22216088898.952103 cv error regularized 4
#########

5.473817582927956e+49 train error regularized 5
5.525112349975599e+49 cv error regularized 5
#########

17885424883.04278 test error regularized 1
17339211427.558002 test error regularized 2
17617027610.07154 test error regularized 3
17647770400.623753 test error regularized 4
5.463107901023058e+49 test error regularized 5