Id	Sq. Ft	Lot	Beds	Baths	Year	Price	Prediction W=[10,1,1,1,1] b=10,000	MSE
1	1826	19378	4	2.5	2005	320000	49649.5	5.22778E+11
2	1050	7500	2	2	2004	185000	30008	
3	1276	12209	3	2.5	2001	215000	36975.5	
4	1040	7658	2	2	2005	319900	30067	
5	1535	4500	2	2	1998	164000	31852	
6	1535	13704	3	2	2001	205000	41060	
7	1040	11143	4	3	2004	340000	33554	
8	1370	13005	4	2.5	1980	260000	38691.5	
9	2036	10207	3	3	2007	227875	42580	
10	2899	13682	3	3.5	2006	438780	54684.5	

Part 3: The reason why squared in necessary in evaluating the performance of a model is to measure the distance from the actual value, so if its negative, it is irrelevant. Another way to do this is to take the absolute value, which would make all the numbers positive

Part 4: I believe the best training model is #1, it did not overshoot the lowest mean square error, and was more efficient in reaching that answer than training #2, training number 3 did not seem to get any better and oscillated between 2 MSE values

Part 5: The reason that the model is perfectly accurate on only the provided data set is because the model is overfit for the training data, and when presented with new data it is too specialized to be correct in its prediction.

Part 6: In order to fix the overfitting problem, we can employ a method discussed in class, Early stopping. If we stop the algorithm before it makes perfectly tailored answers to that one data set, it will be more generalized and be able to apply its knowledge to different sets of data.

Id	Sq. Ft	Lot	Beds	Baths	Year	Price	Prediction W=[10,1,1,1,1] b=10,000	MSE
1	1826	19378	4	2.5	2005	320000	=10*B2+SUM(C2:F2)+10000	=((SUM(G2:G11)-SUM(H2:H11))^2)/10
2	1050	7500	2	2	2004	185000	=10*B3+SUM(C3:F3)+10000	
3	1276	12209	3	2.5	2001	215000	=10*B4+SUM(C4:F4)+10000	
4	1040	7658	2	2	2005	319900	=10*B5+SUM(C5:F5)+10000	
5	1535	4500	2	2	1998	164000	=10*B6+SUM(C6:F6)+10000	
6	1535	13704	3	2	2001	205000	=10*B7+SUM(C7:F7)+10000	
7	1040	11143	4	3	2004	340000	=10*B8+SUM(C8:F8)+10000	
8	1370	13005	4	2.5	1980	260000	=10*B9+SUM(C9:F9)+10000	
9	2036	10207	3	3	2007	227875	=10*B10+SUM(C10:F10)+10000	
10	2899	13682	3	3.5	2006	438780	=10*B11+SUM(C11:F11)+10000	

Part 3: The reason why squared in necessary in evaluating the performance of a model is to measure the distance from the actual value, so if its negative, it is irrelevant. Another way to do this is to take the absolute value, which would make all the numbers positive

Part 4: I believe the best training model is #1, it did not overshoot the lowest mean square error, and was more efficient in reaching that answer than training #2, training number 3 did not seem to get any better and oscillated between 2 MSE values

Part 5: The reason that the model is perfectly accurate on only the provided data set is because the model is overfit for the training data, and when presented with new data it is too specialized to be correct in its prediction.

Part 6: In order to fix the overfitting problem, we can employ a method discussed in class, Early stopping. If we stop the algorithm before it makes perfectly tailored answers to that one data set, it will be more generalized and be able to apply its knowledge to different sets of data.

Advanced Programming ECE 241 Project 3 Aidan Chin

Part 2

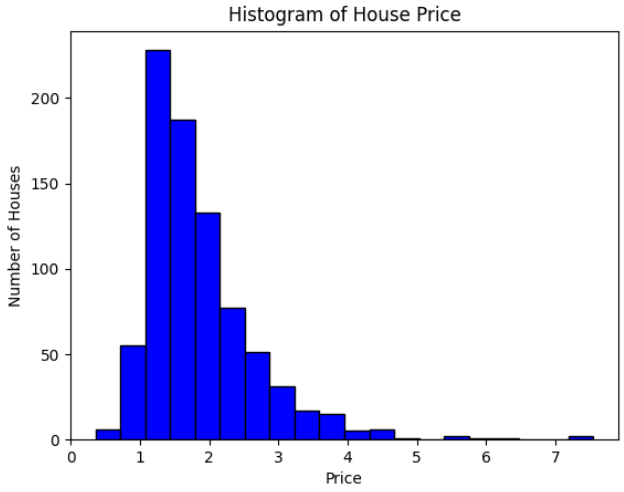
2. Minimum value for price: 0.35311

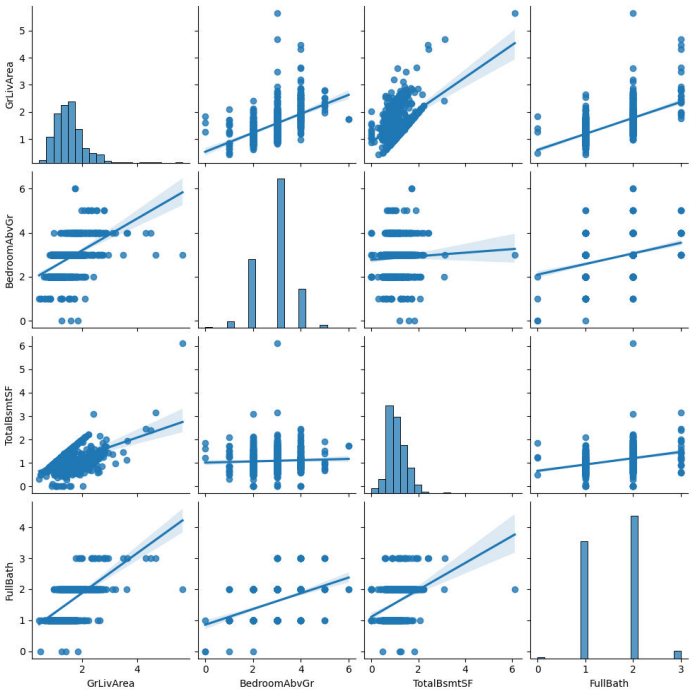
   Mean value for price: 1.8618897432762838

   Maximum value for price: 7.55

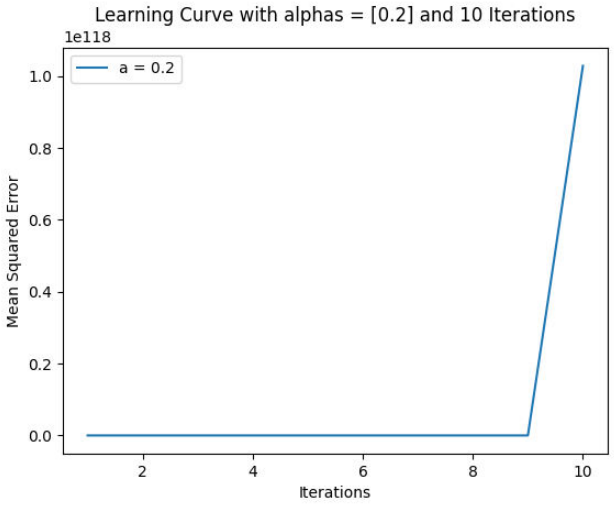
   Standard deviation for price: 0.8242982253562076

   Number of houses: 817

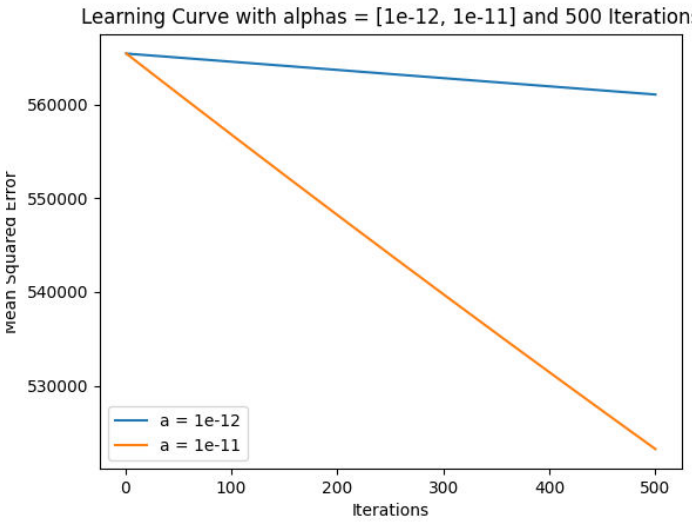
3. 

4. 

From the findings in this plot, the highest correlations are from bedroomabvgr for all across the board and fullbath all across the board. This suggests that there is redundant data. I would suggest removing these 2 from the dataset to improve learning time without compromising accuracy.

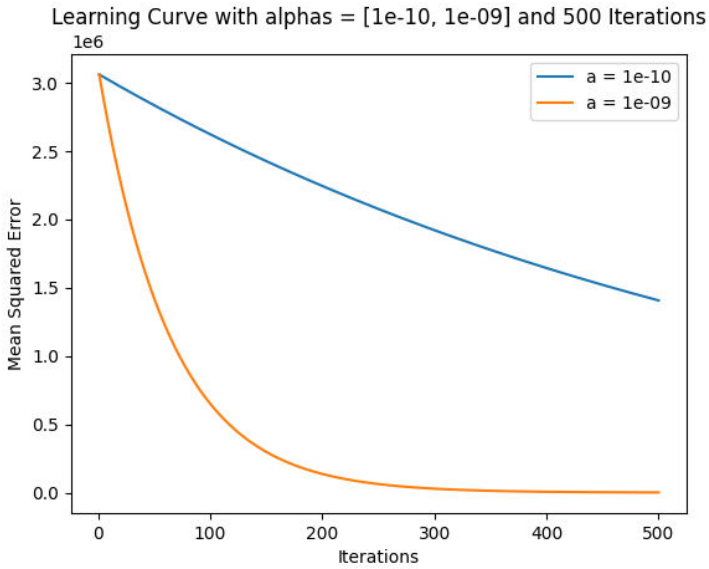
10. 

with the a value at .2, the MSE increases incredibly fast, it goes from 474105.3567108305 at 1 iteration 1.2664088209330762e+18 at just 2 and 2.60294485924787e+117 at 10. The reason this happens is because when the algorithm goes to correct the weights, the a value lets the weights change incredibly fast, not giving the algorithm a chance to converge on a solution, it overshoots too far every time it corrects itself.



11.

12. with the a value at 10**-12 the learning rate is slower than 10**-11 so 10**-11 converges faster, using either of these values will not usually converge before the iteration limit is reached, I propose it would be better to use 10**-9 or 10**-10, thus I ran the algorithm with my values and got a nicer looking graph



13. The MSE using a = 10**-11 during the training ended at 523232.01354840066 and the test MSE was 521826.97919621 and in the few times I tried it, the test MSE was consistently a little less than the training MSE but I predict that will not hold true in the future and it will average out to be about the same. This is because the training has not had time to overfit, and is still learning so the test and training set will have minimal difference in the eyes of the algorithm.

```
PS C:\Users\Aidan\OneDrive\Documents\College\sophmore yr\Advanced Prog\Proj3> & C:/Users/Aidan/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/Aidan/OneDrive/Documents/College/sophmore yr/Advanced Prog/Proj3
/project3.py"
Minimum value for price: 0.35311
Mean value for price: 1.8618897432762838
Maximum value for price: 7.55
Standard deviation for price: 0.8242982253562076
Number of houses: 817
final MSE against training set:  57676.87448675611
test MSE:  56931.359815831274
```